SIEMENS

COMOS

Automation
COMOS automation interfaces

Operating Manual

04/2012
A5E03778189-02

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE

indicates that an unintended result or situation can occur if the relevant information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E03778189-02
℗ 05/2012 Technical data subject to change

Table of contents

Trademarks

1

Trademarks

Registered trademark: COMOS®

SPI 2

2.1 Aim

Aim

The SPI interface imports an Access database that contains a "Smart Plant Instrumentation" (SPI) project. This requires the following workflow to be carried out:

Data in SPI > Import SPI data into Access > Import Access DB into COMOS

No work on the content of the SPI project needs to be done in advance. The data is prepared during the import into COMOS.

Scope of the interface

The import process takes into account the following information (among other things):

- Structures (unit structure, location structure)
- Positions
- Logical components (loops, functions, device requests)
- Electrical components (terminal strips, terminals, cables, wires)
- Control components (PLC, DCS)

The SPI interface is mainly used within the context of EI&C engineering.

The following are not imported:

- User-defined SPI data types ("UDF fields")
- SPI-specific data types ("UID fields")

Interface flexibility

The COMOS administrator defines an assignment between the information from SPI and the COMOS data. Since COMOS data structures are more flexible than SPI data structures, the administrator can freely define the target structure in COMOS.

Changes to the COMOS data can be taken into account in advance. For this purpose, the administrator defines standard assignments in the "Standard object" fields in addition to the concrete assignment. If the COMOS data has changed and a concrete assignment cannot be found during the import process, the standard assignment is used and the import continues.

2.2 Basic principle of importing

Basic principle of exporting from SPI to Access

An ODBC driver controls the process of exporting from SPI to Access. ODBC drivers have their own versions. Switching to a different ODBC driver may change the data that is exported from SPI.

The COMOS installation does not contain any ODBC drivers for SPI.

The COMOS SPI interface does not include an Access license.

Importing an SPI project into Access

Please refer to the Access documentation for information on importing an external data source into an empty Access DB using an ODBC driver. Make sure you adhere to the following rules:

- Import the source data into the database.
- Import all objects.
- Access 2003 and Access 2007 are supported.

2.3 Setting up the ODBC data source

Procedure

To set up an ODBC data source, proceed as follows:

1. Open the menu "Start > Settings > Control Panel".
2. Open "Administration".
3. Open "Data Sources (ODBC)".
4. Select the "System DSN" tab.

   Windows 7: Select the "User DSN" tab.
5. Click "Add" and select an ODBC driver.

   Please refer to the SPI documentation to find out which ODBC driver you need.

   Example: For "Sybase DB 7", the ODBC driver "SQL Anywhere 9" may be used.
6. Click "Finish".

   The names of the next window that appears and the tabs in it will depend on which ODBC driver you select. The steps below use the example of the "SQL Anywhere 9" driver.
7. Select the "ODBC" tab and enter a name for the data source.
8. Select the "Login" tab.

9. Select the "Supply user ID and password" option.

   You can find out your user ID and password from your SPI administrator.

   For example, the user name may be "IN_DBAMN" and the password "IN_DBAMN".

10. Select the "Database" tab.
11. Click "Browse" and select a database.

    The database may be located in the in the "Instrumentation" subfolder of the SPI installation folder, for example.

2.4 Opening the SPI interface and Access DB

To open the "SPI interface" plugin, proceed as follows:

1. Open a project.
2. From the COMOS menu, select the "Plugins > Automation > SPI interface" command.
3. Select the "Import > Import" tab.
4. In the "File" field, select the Access DB in which the SPI project is stored.
5. Select an SPI project in the "Project name" list.

2.5 Overview of the import process

Import process sequence

The SPI information is imported in the following sequence:

1. Hierarchy elements

   Units and locations
2. Positions
3. Functions, panels, and measuring devices

   Functions are created below the positions.

   Panels and measuring devices are created below the locations.
4. Terminal strips

   Terminal strips are created below the panels or measuring devices.
5. PLC/DCS structures

   Racks, cards, and channels
6. Cables, stranding and wires

7. Terminals/connectors

   SPI contains "terminals" which may serve as both terminals and connectors.

8. Connections between the connectors

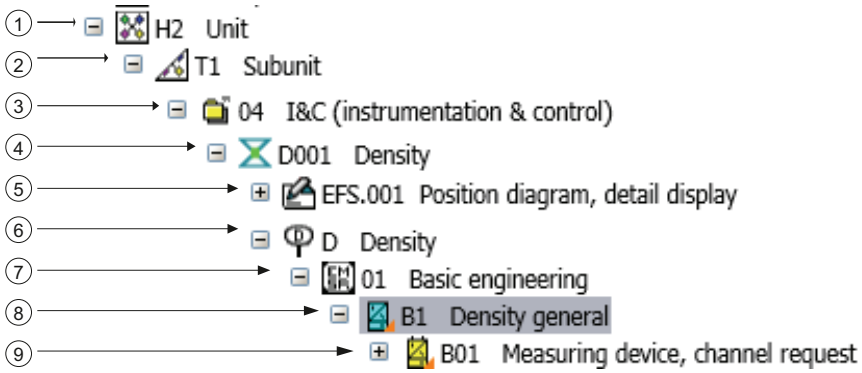
   Auxiliary connectors are created if necessary.

This process is described according to COMOS terminology. See also section Reference: Fixed mapping rules (Page 24).

2.6 Notes on the import steps

Hierarchy of all objects on the "Units" tab

The following structure is created during the import process:



1	Unit
2	Part unit
3	Folder (optional; for the technique "category folder", for example)
4	Position or loop
5	Position diagram
6	Function
7	Folder ("Substructure")
8	Measuring device
9	Control component ("Structure element") or measuring device request

This process is described according to COMOS terminology. See also section Reference: Fixed mapping rules (Page 24).

Importing SPI-specific standard elements

The "Empty" and "Default" SPI elements are imported for reasons of completeness.

These elements do not usually contain any project-specific information.

Importing a function with multiple devices

Definition:

- The information labeled "Component" in SPI is referred to as "Function" here.
- The information labeled "Panel, category 4" in SPI is referred to as "Measuring device" here.

If a function has multiple devices, the following applies:

- All the devices are created below the functions.
- The devices receive the connector objects.

Importing a device that is assigned to multiple functions

If a device has multiple signals, this device has to be assigned to multiple functions. The following applies in this case:

- All the functions found are imported.
- All connector objects are assigned to the device. The device is assigned to the first function that is found.

Importing functions without devices

If a function for import does not have any devices, the function is imported and does not have any connectors.

Importing SPI connectors

Connectors for devices and terminals are treated in the same way in SPI.

For connectors of the panel categories 1, 2, 3 and 5, 6, 7, the following applies: Terminals are created below the terminal strips and the connectors are grouped below the terminals.

The following applies to the connectors of panel category 4: The panel is treated as a measuring device and a measuring sensor is created below the measuring device. The connectors are created below the measuring sensor.

The connectors are separated logically during the import process. Device connectors are created below the devices in the form of connector objects without terminal strips. Terminals are created below terminal strips.

The following "Cable" information is set as attributes:

- "JUMPERS"

  "JUMPERS are imported as wire jumpers with the following versions:

  – Internal Connector (I): Index = "@B.VBI"
  – External Connector (O): Index = "@B.VBE"
- "CROSS WIRE"
  – Wire color

2.7 Preparing to import objects (assignment)

2.7.1 Information on the "Hierarchy objects" tab

Initializing the tab

Definition: The term "hierarchy" is used in SPI; this corresponds to the unit structure in COMOS.

The button "Load hierarchy information from database" creates a table. This table lists the hierarchies of the SPI project.

Information on assignment

Work in SPI is carried out using fixed hierarchies. Here is an example based on 3 levels:

- Top level

  This level does not have an owner. In COMOS this corresponds to the top object on the "Units" tab. However, it is not technically essential to use such an object in COMOS.

- Second level

  This level is created below the top level in the hierarchy. During the import process, additional levels can be created between the top level and second level by using defaults in the COMOS base data. You cannot create additional levels on the "Hierarchy objects" tab. The visible entries on this tab (and therefore the minimum number of levels to be created) are defined by the database to be imported.

- Third level

  This level is created below the second level in the hierarchy. During the import process, additional levels between the second level and third level can be created using COMOS templates.

Example

Some examples of SPI table names provide a clear overview of how this system works:

- Top level: DEMO_PLANT
- Second level: DEMO_PLANT_AREA
- Third level: DEMO_PLANT_AREA_UNIT.

The following assignment may be used:

Level 1 (Plant)	"@02 General objects > 010 > 1 > PID > 01 Plant"
Level 2 (Area)	"@02 General objects > 010 > 1 > PID > 02 Unit"
Level 3 (Unit)	"@02 General objects > 010 > 1 > PID > 03 Part unit"

2.7.2 Information on the "Structure objects" tab

Information on assignment

Definition: The term "structure" is used in SPI; this corresponds to the location structure in COMOS.

COMOS contains two different tabs called "Units" and "Locations", each of which has a separate hierarchy.

In SPI, however, there is no clear distinction between the unit tree and location tree. From the user's perspective, all of the nodes/folders are located in the same tree. During the import process, a location is created in COMOS for each top-level unit that is created.

Other lower-level location structures are derived from the SPI project data. Standard projects contain the following folders, for example:

- Component Locations
- Panel Locations
- Panels by category

  SPI features 7 types (or "categories") of panels. A subfolder is prepared for each category.

Example

The following assignment may be used:

Location	"02 General objects > 060 > 1 > 00 Plant"
Device location	"02 General objects > 060 > 1 > 04 Cabinet"
Function location	A base object you define yourself in "02 General objects > 060 > 1 Base objects"
Folder	"02 General objects > 010 > 2 > PID C > 99 Miscellaneous"

2.7.3 Information on the "Positions" tab

Initializing the tab

The button "Load loop information from database" creates a table. This table lists the positions of the SPI project.

Information on assignment

For all entries without an appropriate individual assignment, the "Standard object" field is used. If the position has multiple position diagrams in the "Standard object" field, you must assign a position diagram in the "Document" list.

You can use the "Start object" field to set the start object for automatic assignment. If COMOS finds a position with an appropriate position code under the start object, assignment between the SPI position and COMOS position is carried out automatically.

You also have the option of assigning the positions manually, on an individual basis. The SPI positions largely conform to the DIN/IEC standard for positions. In COMOS, a limited library is also available for this standard.

Example

The following assignment may be used:

Start object	"@03 Structures > BAS > IEC > P Positions"
Standard object	"@03 Structures > BAS > IEC > P > 0 Undefined"
Document	<Position diagram> Here you see the interactive reports that are underneath the object that is set in the "Standard object" field.
SPI process name	Assignment of COMOS positions in "@03 Structures > BAS > IEC > P Positions"

2.7.4 Information on the "Functions" tab

Initializing the tab

The button "Load function information from database" creates a table. This table lists the functions of the SPI project.

Information on assignment

If no individual assignment can be found for an entry, the assignment in the "Standard function" and "Standard measuring device" fields is used.

You can use the "Start object for searching functions" and "Start object for searching instruments" fields to set the start objects for automatic assignment. If COMOS finds a function with an appropriate function code below the start object, the SPI function is automatically assigned to the COMOS function.

You also have the option of assigning the functions manually, on an individual basis. For this purpose, you need to set the COMOS measurement functions in the left-hand area of the table. You can select multiple rows at once and implement a common assignment by means of a single drag&drop operation.

If the COMOS measuring function has a folder into which the devices are sorted, then this folder is displayed in the "Substructure" column.

The right-hand area of the table is where the COMOS devices are set. In the COMOS standard model, the devices associated with a measuring function are stored in the "01 Basic Engineering" folder.

Example

The following assignment may be used:

Start object for searching functions	"@03 Structures > BAS > IEC > F Functions"
Start object for searching instruments	"@01 Material > EIC > 01 > 170 Measurements"
Standard function	<Dependent on project>
Standard measuring device	Base object you define yourself in "@01 Material > EIC > 01 > 170 > Z04 General (miscellaneous) equipment"

An entry in the assignment table may be processed as follows:

ID	Function type	Description
27	FR	LOCAL FLOW RECORDER
COMOS function:		Substructure (entered automatically):
"@03 Structures > BAS > IEC > P > F > F Flow"		"01 Basic Engineering"
COMOS measuring device:		
"@03 Structures > BAS > IEC > P > F > F Flow > 01 Basic Engineering > A > A > F > F5 General flow sensor"		

2.7.5 Information on the "Devices" tab

Information on assignment

The devices are assigned on this tab. Category "4" SPI panels and SPI components count as devices.

For all entries without an appropriate individual assignment, the "Standard object" field is used.

If the "Search object", "Manufacturer attribute", and "Model attribute" fields are assigned, COMOS can also assign the objects using a manufacturer device search.

In the COMOS standard model, these objects are created on the "Locations" tab.

Where possible, the attribute "Sys.Areaname" is set for the devices during the import process.

The devices are located below the functions and contain the connectors.

Example

The following assignment may be used:

Terminal box	"@02 General objects > 060 > 1 > 31 Field distributor"
Marshalling	"@02 General objects > 060 > 1 > 11 Marshalling"
Control cabinet	"@02 General objects > 060 > 1 > 51 Controller (PLC)"
DCS	"@01 General objects > EIC > 03 > COMOS > C > ST Station"

PLC	"@02 General objects > 060 > 1 > 51 Controller (PLC)"
Telecommunication device	Base object of a control cabinet; e.g. "@02 General objects > 060 > 1 > 04 Cabinet"

You can find an example for storing manufacturer device data in COMOS in the chapter titled Reference: Manufacturer device data in COMOS (Page 25).

2.7.6 Information on the "Terminal strips" tab

Information on assignment

The standard terminal strip is assigned on this tab.

Where possible, the attribute "Sys.Areaname" is set during the import process.

Example

The following assignment may be used:

Standard terminal strip	"@03 Structures > EIC > 300 > 01 > 01 Terminal strip, general"

2.7.7 Information on the "Terminals" tab

Information on assignment

The standard terminals are assigned on this tab.

For all entries without an appropriate individual assignment, the "Standard object" field is used.

If the "Search object", "Manufacturer attribute", and "Model attribute" fields are assigned, COMOS can also assign the objects using a manufacturer device search.

Example

The following assignment may be used:

Terminal	"@01 Material > EIC > 01 > 750 > X > C > A > 01 Lead-through terminal, 2-conductors"
Terminal (N)	"@01 Material > EIC > 01 > 750 > X > C > A > 02 Lead-through terminal, 2-conductors, N"

Terminal (PE)	"@01 Material > EIC > 01 > 750 > X > C > A > 03 Lead-through terminal, 2-conductors, PE"
Terminal (H)	"@01 Material > EIC > 02 > 750 > 01 > 06 Shield terminal, 2 conductors"

You can find an example for storing manufacturer device data in COMOS in the chapter titled Reference: Manufacturer device data in COMOS (Page 25).

2.7.8 Information on the "Cables" tab

Information on assignment

The standard cables are assigned on this tab.

For all entries without an appropriate individual assignment, the "Standard object" field is used.

If the "Search object", "Manufacturer attribute", and "Model attribute" fields are assigned, COMOS can also assign the objects using a manufacturer device search.

In SPI, a cable can be created below a function and a device as well as below a terminal strip. During the import process, all cables receive a device or measuring device as the owner.

The SPI cables "CROSS WIRE" and "JUMPER" are not created as a cable object during the import process. Instead, this information is created as an attribute at the connector object.

Example

The following assignment may be used:

Standard object	"@03 Structures > EIC > 310 > 04 > 0000 Conductor, unspecified"
Search object	"@03 Structures > EIC > 310 > 04 Cables/conductors"

You can find an example for storing manufacturer device data in COMOS in the chapter titled Reference: Manufacturer device data in COMOS (Page 25).

2.7.9 Information on the "Wires" tab

Information on assignment

The standard wires are assigned on this tab.

The following structure must be used in SPI: Cables > stranding > wires.

This means that both the "Standard wires" and "Standard stranding" fields have to be assigned in order to import wires.

Example

The following assignment may be used:

Standard wires	"@01 Material > EIC > 02 > 850 > ELM01 Wire"
Standard shield	"@01 Material > EIC > 02 > 850 > SH Shield"
Standard stranding	"@01 Material > EIC > 02 > 850 > VS Stranding"

2.7.10 Information on the "PLC/DCS" tab

Information on assignment

The standard control components are assigned on this tab.

Example

The following assignment may be used:

Rack	"@01 Material > EIC > 03 > COMOS > A > 00 > 01 Module rack, 10 slots" or "@02 General objects > 060 > 1 > 05 Rack"
Card	"@01 Material > EIC > 03 > COMOS > A > 04 > NN Module (unspecified") or "@02 General objects > 060 > 1 > 06 Slot"
Channel	- (Dependent on task)

You can find an example for storing manufacturer device data in COMOS in the chapter titled Reference: Manufacturer device data in COMOS (Page 25).

2.8 Preparing to import attributes (assignment)

2.8.1 Information on the attribute tabs

Information on assignment

The SPI data fields are assigned to COMOS attributes on these tabs. Only assigned data fields are imported.

However, assigning SPI data fields to COMOS attributes is optional; all the other elements of the import can still be used, even if nothing has been assigned to the attributes.

The "Min" and "Max" information is stored in separate fields in SPI. In COMOS, this information is part of an attribute object. For this reason, you only assign a COMOS object once; the "Min" and "Max" information is assigned automatically.

2.8.2 Assigning attributes

Procedure

1. Click "...load from the database".

   A table appears on the tab. A row is created for each Smart Plant data field.
2. Use drag&drop to move an attribute from the Navigator to the "COMOS attribute" column.
   - "Description of COMOS attribute" column: The value in this column is entered automatically.
   - "Min/Max" column: The value in this column is entered automatically.

2.9 Conducting the import

Controlling the import

- "Create working layer"

  Creates a working layer below the current working layer during the import process.
- "Cancel import"

  This button appears after the import has started and is used to abort the import.

Starting the import

Click "Start import".

Checking the import

The import status is displayed in the COMOS status bar.

- "Display log file"

  Displays the log file in a standard text tool (frequently Notepad).

The log file is stored in the following location:
`C:\Documents and Settings\<User>\Application Data\Comos_Industry_Solutions\SPI_Import`

The assignments in the SPI interface are stored in the following profile object:

- "@System > @Profiles > <User>"

or

- "@System > @Profiles > @AllUsers > Comos.IC.SPI"

2.10 Reference: Fixed mapping rules

Logical assignment of base objects

SPI	COMOS
Loop	Position
Component	Function
Panel (Cat. 4)	Measuring device
Panel	Location
Strip	Terminal strip
Terminal	Terminal object or connector object
Cable	Cable
Wire	Wire
Location	Location
Panel (Cat. 5)	Station
Cabinet_Rack	Rack
Rack_Position	Slot (Not an object to be created. Instead, the rack card receives the slot name.)
Apparatus	Card/rack
Channel	Channel

Logical assignment of SPI panel categories

Category	SPI	COMOS
1	Junction box	Field panel
2	Marshaling cabinet	Marshaling panel (field)
3	Control cabinet	Marshaling panel (control)
4	Panel	Measuring device (field device)
5	DCS	Control component
6	PLC	Control
7	Telecom device	Cabinet

2.11 Reference: Manufacturer device data in COMOS

Storage in COMOS

Example of a base object under which manufacturer devices are grouped: "@01 Material > EIC > 01 Objects available for order".

Manufacturer specification	"HSD.M003 Manufacturer"
Model specification	"HSD.M035 Order number"

Additional example of an attribute with specifications on the article: "HSD order data" tab, "M001 article number" attribute.

Automation PCS 7 Interface 3

3.1 Objective and scope of performance

Objective

The COMOS plugin "Plugins > Automation > PCS 7 interface" offers a bidirectional interface for the exchange of engineering data with SIMATIC PCS 7. Following initialization of the COMOS project and the PCS 7 project, you can work in both programs alternately.

The objective is a "digital unit" through integrated engineering with COMOS and SIMATIC PCS 7. In the digital unit, unit planners and unit operators work without interruption on the same database and so generate continuous engineering.

Performance scope of the interface

- Importing control module types from PCS 7 to COMOS
- Bidirectional import/export of control modules and parameters, messages and signals
- Bidirectional import/export of signal connections
  - Implementation of channels and signals
  - Block logic in function diagrams
- Bidirectional import/export of hardware including symbol tables
- Bidirectional import/export of redundant CPU hardware including symbol tables

  The following applies to the import of redundant CPU hardware: the redundant hardware is only exported to the Automation Tree.
- Bidirectional import/export of hierarchy folders (structure of the "Units" tab)
- Export of interlock logic to function diagrams (IEC blocks connected across documents) from COMOS to PCS 7

  Only blocks with at least one interconnected input are exported.

Exceptions

The following engineering data is not considered:

- No import of the graphical arrangements of the hardware configuration in HW Config of STEP 7.
- No import of the graphical display of CFC and functional diagrams.

See also

Software requirements (Page 28)

3.2 Software requirements

The following software must already have been installed:

- COMOS 10.x
- Latest version of the Automation Interface for PCS 7 V8

  The Automation Interface is installed with the COMOS CD browser. You can find additional information on this topic in the COMOS system administration documentation "Installation", keyword "COMOS CD browser".
- PCS 7 V8.0
- CFC V8.0 SP1
- COMOS-DB V 10.x

Only the versions listed above are permitted.

See also

Terminology (Page 28)

3.3 Terminology

Important terminology

- Automation Interface

  The Automation Interface is an independent, standardized interface for the exchange of engineering data. The user interface for the Automation Interface has the dialog title "Data transfer - Generate/Import".
- Signal

  In PCS 7: The designation "signal" is only used when the signal is connected to a hardware address.

  In COMOS: Signals are used regardless of assignment to hardware.

Changes to the terminology

Old	New
Automation view	Automation tree
eBlock	Task/Engineering task
Process tag	COMOS-specific object: Process tag, COMOS process tag or function. PCS 7-specific object: Control module (CM).
Process tag types	Control module types
Process tag library	Control modules library

Old	New
"PT" menu	"Automation" menu
"AdvES" tab	"Function specifications" tab
"Revise" function	"Reengineering" function

Synonyms

The following terms each have the same meaning:

- Interface module, interface module
- Block message, message

Terms with overlapping meaning

Symbolic name	Symbolic address
A symbolic name is a placeholder for a unique designation to be assigned at a later time, for example, an address. Symbolic names are often spoken names.	A symbolic address is a symbolic name with the following special characteristic: A symbolic address is derived from the plant hierarchy and therefore depends on the hardware.

See also

Information on the identification of SIMATIC components (Page 29)

3.4 Information on the identification of SIMATIC components

Objective

All components that are imported or exported through the PCS 7 interface must receive a unique designation both in COMOS and in PCS 7. The data can only be assigned automatically during import or export if they have unique designations.

Exceptions

The identification based on PCS 7-AppID is currently not available.

A comparison of the identification fields of COMOS and PCS 7

The components are identified using the "Name" and "Item number" fields. In COMOS, the properties "Name" and "Item number" are generated according to rules that differ from PCS 7.

	COMOS rule	PCS 7 rule (as per HW Config)
Name	Sequential number	Number according to slot rules Slot number of the module in the rack or distributed I/O. See the description on "Basics of slot rules in PCS 7" below.
Item number	Freely selectable item number	Machine-readable serial number (MLFB) of the manufacturer Example: "6ES7 417-4XT05-0AB0".

The basic principle is as follows: The objects are initially created based on the COMOS rules in COMOS. The objects then have to be renamed according to the PCS 7 rules so that they can be exported.

Basics of slot rules in PCS 7

Both the slot rules of the rack and the connection rules of the bus system are important for the identification of a component in PCS 7. As a general rule, a SIMATIC controller is structured as follows:

- Structure of the rack according to the slot rules of HW Config: Bus address

  The power supply, the CPU and the input/output modules are plugged into the rack. The bus address must be entered in the COMOS object and be unique on the bus.

  Example: A "1" is entered for the power supply.

- Connections according to the connection rules of the bus system: Addresses of inputs and outputs

  The bus system is connected to the CPU or a communication processor (CP). The addresses are managed in blocks of 8 and written with the notation "Byte.BIT". The value ranges from 0 to 7. The y position is incremented first, then the x position.

  Example: The addresses for a digital input module with 24 inputs range from 0.0 to 3.7.

Standard check when exporting from COMOS to PCS 7

When exporting the hardware configuration:

- The configured components are identified using the "Name" and "Item number" attributes.

Note

Checking the slot rules of HW Config

The content of the "Name" attribute is initially checked in the Automation Interface. In the event of an error, the following message appears:

"The value of the "Name" attribute violates a slot rule of HW Config."

Standard check when importing from PCS 7 to COMOS

In the case of reengineering, the following applies to importing the hardware configuration:

- The configured components are identified using the "Item number" attribute and assigned to the base objects in COMOS.

See also

Overview of the standard work procedure (Page 31)

3.5 Overview of the standard work procedure

Perform at least the following steps to exchange engineering data between COMOS and SIMATIC PCS 7:

Preparation by the administrator

1. Prepare PCS 7

   See chapter Requirements in PCS 7 (Page 33).
2. Prepare COMOS
   - Prepare COMOS base data: Classify plant hierarchy

     See chapter Preparing base objects as a technological hierarchy (Page 34).
   - Prepare a COMOS engineering project

     See chapter Initializing a COMOS engineering project (Page 37).
3. Manage control module types in COMOS
   - Import control module types in COMOS.

     See chapter Importing individual control unit types (Page 41).
   - Create templates for control module types.

     See chapter Creating templates of individual control unit types (Page 43).
   - Configure a mapping table for the control module types.

     See chapter Configuring the mapping table of the signal designations (Page 72).

Engineering by the COMOS user

1. Hardware engineering in the "Locations" tab

   See chapter Hardware engineering in the Navigator (Page 46).
   - Configure the control cabinet
   - Create and configure the CPU
   - Create and configure an IO module
2. Software engineering (generate plant hierarchy) in the "Units" tab

   See chapter Software engineering in the Navigator (Page 56).
   - Create COMOS process tags
   - Signal engineering
   - Implement channels

     See chapter Linking hardware and software: Implementing channels (Page 60).

     This step links the hardware engineering and the software engineering
3. Assign control modules to COMOS process tags

   Editing software engineering using engineering tasks (Page 61)
4. Create or update symbol table

   See chapter Using symbol tables (Page 66).
5. Create interlock logic on function diagrams

   See chapter Engineering in the function diagram (Page 68).

Final work by the administrator or super user:

1. Link COMOS attributes and PCS 7 attributes

   See chapter Linking attributes from COMOS and PCS 7 (Page 72).
2. Generate PCS 7 project and export from COMOS to PCS 7

   See chapter Exporting from COMOS to PCS 7 (Page 77).
3. Perform import from PCS 7 to COMOS

   See chapter Importing from PCS 7 to COMOS (Page 83).
4. Optional: Reengineering of hardware and software

   See chapter Reengineering (Engineering in PCS 7) (Page 89).

3.6 Requirements in PCS 7

3.6.1 Requirements in PCS 7

Requirements for control module types

To import control module types from a PCS 7 project, the following conditions must be met:

- The PCS 7 project is part of a multiproject
- The multiproject contains a master data library
- The master data library contains control module types

Requirements for messages

To import messages from a PCS 7 project, the following conditions must be met:

- The message must be assigned to the block of a control module in PCS 7.
- The message must be assigned to the block of a control module or the block variable of a control module at the plant interface in PCS 7.

Requirements for optional blocks (variants of control module types)

To import optional blocks from a PCS 7 project, the following conditions must be met:

- The block must be optionally assigned to a control module in PCS 7.
- The block must be assigned to a plant connection at the plant interface in PCS 7.
- The "Optional" checkbox must be set for this block.

Requirements for user-defined attributes

To import user-defined attributes from a PCS 7 project, the following conditions must be met:

- The user-defined attribute must be assigned to a plant connection at the plant interface in PCS 7.
- One of the following attributes is used:
  - Type
  - Name
  - Comment
  - Symbol
  - Cycle time

Requirements for interlock logic on function diagrams

To import interlock logic of blocks from a PCS 7 project, the following conditions must be met:

- The hardware driver block and its connections must be assigned to a plant connection at the plant interface in PCS 7.

See also

Preparing COMOS (Page 34)

3.7 Preparing COMOS

3.7.1 Preparing base objects as a technological hierarchy

Requirement

- A PCS 7 project has been prepared.

  See section Requirements in PCS 7 (Page 33).

Objective

Preparation of COMOS objects in order for them to be used as a plant hierarchy.

Set classification "PH Plant Hierarchy"

1. On the "Base objects" tab in the base project, navigate to the object that you wish to classify.
2. Open the properties.
3. Select the "System settings" tab.
4. Select the "PH" entry in the "Functional classification" list.
5. If required: perform a classification update in the engineering project.

   You can find additional information on this topic in the "Queries" manual, keyword "Perform classification".

Create "AES020 PCS 7" tab and "AES0022 PCS 7 hierarchy name" attribute

1. On the "Base objects" tab in the base project, navigate to the object that you wish to edit.
2. Open the properties.

3. Open the "Attributes" tab.
4. Generate the "AES020 PCS 7" tab based on the following entry in the attribute catalog:

   "@10 @Y Attribute catalog > EIC > 2 > AES020 PCS 7"
5. Check if the "AES0022 PCS 7 hierarchy name" attribute is available:
   - Open the "AES020 PCS 7" tab.
   - Check if the "AES0022 PCS 7 hierarchy name" attribute is available.

     The "AES0022 PCS 7 hierarchy name" attribute specifies the name of the COMOS object in the plant hierarchy of PCS 7. If this attribute is empty, then the object will also not be exported.

You can find additional information on this topic in the "Database Design Guide" manual, keyword "Add tabs to objects".

Prepare "OnMove" script block

A script of the PCS 7 library must be called in the "`OnMove`" script block:

```
Project.Workset.Lib.CallScriptLib "@02|200|AES|S|00","Move",object
```

Preparing engineering tasks

The following engineering tasks must be available:

- Copy and paste control module
- Assigned PLC

Examples of a plant hierarchy in the COMOS DB

In the COMOS DB, the following base objects are already prepared as a plant hierarchy:

- Unit: "@02 General objects > 010 > 1 > PID > 02 Unit"
- Part unit: "@02 General objects > 010 > 1 > PID > 03 Part unit"
- Positions: "@02 General objects > 020 > 00 Positions to DIN/IEC"

See also

Assigning a plant hierarchy in COMOS (Page 36)

Notes on the plant hierarchy (Page 56)

Adapting the plant hierarchy (Page 93)

3.7.2 Assigning a plant hierarchy in COMOS

Requirement

- The base objects have been prepared.

  See section Preparing base objects as a technological hierarchy (Page 34).

Assignment in the properties of the project

1. Open the "Open project" tab.
2. Open the properties of the project.
3. Open the " Options > PCS 7 interface" tab.
4. Assign the COMOS base objects to fields "Level 1" to "Level 8".

Default hierarchy

The default case can work even without assignment if the base objects were prepared as a plant hierarchy. A default assignment is prepared as follows:

"@03 Structures > AES > 300 > 01 Hierarchy folder"

These hierarchy folders are available here, for example:

"@02 General objects > 010 > 1 > AES > 01 Hierarchy folder"

Plant hierarchy including COMOS objects

For the default case, see Notes on the plant hierarchy (Page 56).

See also

Opening a COMOS engineering project (Page 36)

3.7.3 Opening a COMOS engineering project

Requirement

- The COMOS base data have been prepared.

  See section Preparing base objects as a technological hierarchy (Page 34).
- The COMOS project properties have been prepared.

  See section Assigning a plant hierarchy in COMOS (Page 36).

Prepared engineering object

The documentation is based on the following sample project:

1. Select the "File > Open project" command in the menu bar.

   The "Open project" tab opens.

2. Open the project "COMOS_PCS 7 Sample project PCS 7".

You can find additional information on this topic in the "COMOS Platform Getting Started", manual, keyword "Open project and working layer".

Creating your own engineering project

Use the following project structure:

- "@J Project > @G Project settings, general"

See also

Initializing a COMOS engineering project (Page 37)

3.7.4 Initializing a COMOS engineering project

Requirement

- A COMOS engineering project is open.

  See section Opening a COMOS engineering project (Page 36).

Creating the "@AT Automation Tree" object

1. In the COMOS menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.

   The "@AT Automation Tree" object is created automatically in the "Units" tab.

   The "@AT Automation Tree" object uses the following base object: "@System System settings > @D > @Automation > @AT Automation Tree".

Create the "New project" object

1. In the COMOS menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.

   The "@AT Automation Tree > New project" object is created automatically in the "Units" tab.

   The "@AT Automation Tree" object uses the following base object: "@System System settings > @D > @Automation > @Simatic > PRJ Project".

Rename the "New project" object

1. Go to the "Units" tab.
2. Open the structure below "@AT Automation Tree".
3. Open the properties of the "New project" object.
4. Change the entry in the "Name" field.
5. Confirm your entries.

   The change is also automatically applied in the "Automation Tree" tab.

Creating an additional project in the Automation Tree

1. In the COMOS menu, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.
2. Go to the "Generate" tab.
3. Open the list in the "PCS 7 project" column.
4. Select the "New" entry from the list.

   A new project is created below "@AT Automation Tree" in the "Units" tab. You can then specify the project name and project path in the "Create new project" dialog. The new project is also automatically displayed in the "Automation Tree" tab.

Delete project

1. Go to the "Units" tab.
2. Open the structure below "@AT Automation Tree".
3. Select a project.
4. Select "Delete" from the context menu.

   The change is also automatically applied in the "Automation Tree" tab.

See also

Management of individual control unit types in COMOS (Page 39)

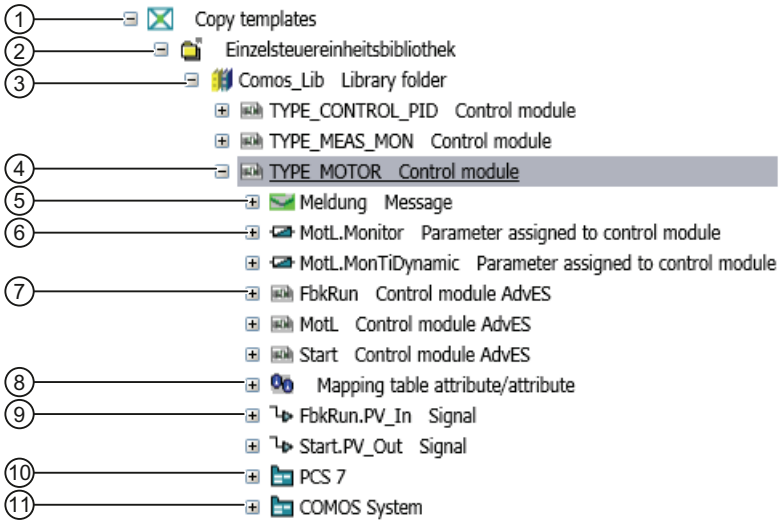
3.8 Management of individual control unit types in COMOS

3.8.1 Structure of control module types

Requirement

- The COMOS engineering project has been initialized.
  See chapter Initializing a COMOS engineering project (Page 37).

Structure



①	COMOS-specific, permanent folder
②	General collecting folder for all templates of the PCS 7 interface
③	A collecting folder from the Automation Tree.
④	Control module type or variants
⑤	Message Messages have the following properties: • They describe events for control modules. • They are classified by the attributes "Message class", "Priority" and "Origin". • The have a freely available "Event" field to describe the event. • The have a freely available "Infotext" field.
⑥	Parameter or block variable of the control module
⑦	Block of the control module
⑧	Mapping table for linking PCS 7-specific attributes with COMOS-specific attributes

⑨	Signal
⑩	PCS 7-specific attributes
⑪	Attributes for controlling the behavior of control modules in COMOS

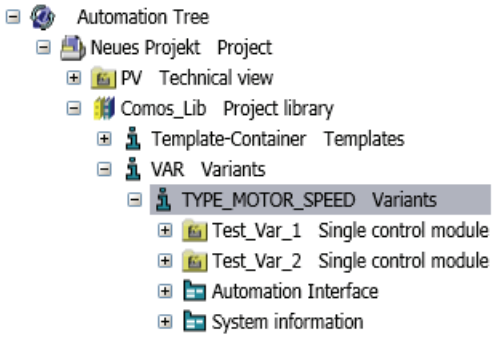
Optional blocks (variants)

Optional blocks are used to define the maximum size and the minimum size of a control module type. The difference between these two states is mapped by the optional blocks. The objective is to define a control module type so that it can be used as a template for the various control module types that are similar in function.

The user decides while creating the templates which optional blocks he wants to use. The blocks available for a control module can no longer be changed.

The optional blocks are managed and identified as follows:

- "Units" tab, "Automation Tree" branch
- <Project> <Collecting folder of the Comos_Lib> < Collecting folder of the variants> <Control module type> <Variants of the control module type>



- The variables, together with all other control modules, are contained on the "Templates" tab under "Comos_Lib library folder"
- Block in a variant of the control module type:

  "Automation Interface" tab, "Optional" attribute

See also

Importing individual control unit types (Page 41)

Managing variants using optional blocks (Page 44)

3.8.2 Importing individual control unit types

Requirement

- The COMOS project has been initialized.

  See section Initializing a COMOS engineering project (Page 37).
- The structure of control module types is clear.

  See section Structure of control module types (Page 39).

Objective

The import of control module types from PCS 7 to COMOS. Control module types cannot be changed in COMOS. Later adjustments of the control module types are also made in PCS 7 and are reimported into COMOS.

Procedure

1. In the COMOS menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.
2. Select the "Import" tab.
3. Select the file type of the import file:
   - The "PCS 7 project" option: s7p
   - The "XML file" option: xml
4. In the "PCS 7 project/library" field, select the source project to be imported.
5. In the "Filter" group, activate the option "Control module types".

   The message "Confirm control module library" appears.
6. Deactivate all other options in the "Filter" group.
7. In the "Automation Tree project/library" list, select the COMOS target to be imported.
8. Click on the "Import" button.

   The "Data transfer - generate/import" dialog opens. See section Information in the "Data transfer - Generate/import" dialog (Page 94).
9. Click on the "Import objects from B to A" button in the "Data transfer - generate/import" dialog.

Result

The library with the control module types is imported into the COMOS project and is visible as follows:

- "Units" tab: "Automation Tree > Project > Comos_lib Project library > Template-Container Templates"
- "Automation Tree" tab: "Project > Comos_lib Project library> Template-Container Templates"

See also

Reimporting individual control unit types (Page 42)

3.8.3 Reimporting individual control unit types

Requirements

- The control module types have been imported in COMOS.

  See section Importing individual control unit types (Page 41).

Objective

The reimport of control module types from PCS 7 to COMOS.

Procedure

1. Follow the same steps as when importing control module types. See section Importing individual control unit types (Page 41).

   Reimporting in COMOS is only possible if identical control module types are used in PCS 7 and COMOS.

2. Please note the following:

NOTICE
Loss of information for the control modules If revised control module types are imported in COMOS, they have an effect on control modules that already exist in COMOS. • If objects or attributes (values) were deleted in the control module type, these objects and attributes are also deleted at the control modules in COMOS. • If values have been changed in the control module type, the previously changed values are retained in the control modules in COMOS.

Result

The library with the control module types is reimported into the COMOS project and is visible as follows:

- "Units" tab: "Automation Tree > Project > Comos_lib Project library > Template-Container Templates"
- "Automation Tree" tab: "Project > Comos_lib Project library> Template-Container Templates"

See also

Creating templates of individual control unit types (Page 43)

3.8.4 Creating templates of individual control unit types

Requirements

- The control module types have been imported from PCS 7 to COMOS.

  See section Reimporting individual control unit types (Page 42).
- The following node is visible in the "Automation Tree" tab: "Project > Comos_Lib Project library> Template-Container Templates". If the project library is missing, the message "Missing ptt library" appears.
- Optional: The variants of the control module types have been created.

  See section Managing variants using optional blocks (Page 44).

Objective

Generating templates from which the control modules can be derived in the "Units" tab.

Creating templates by means of mapping codes

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Create templates" command.

   The "Create templates" tab is opened.
2. Select the "Mapping code" tab.
3. Select an entry in the "PCS 7 Project" list.
4. Click the "Generate" button.

Result

The control module types are created as follows:

- The "Automation Tree" tab

  "Automation Tree > Project > Comos_Lib Project library > Template-Container Templates"

- "Units" tab

  "@Template Copy templates > @PCS 7 Control module library > Comos_Lib Library folder"

See also

Engineering in COMOS (Page 46)

3.8.5 Managing variants using optional blocks

Requirements

- The control module types have been imported or re-imported from PCS 7 to COMOS.

  See section Importing individual control unit types (Page 41).

- The following node is visible in the "Automation Tree" tab: "Project > Comos_Lib Project library> Template-Container Templates". If the project library is missing, the message "Missing ptt library" appears.

Objective

Generating variants for copy templates from which the control modules can be derived in the "Units" tab.

Creating templates with variants

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Create templates" command.

   The "Create templates" tab is opened.

2. Select the "Variants" tab.
3. Select an entry in the "Control module types" column.

4. Check whether the variant is permitted.

   A variant is permitted if:

   – Blocks are listed in the "Control module type variants" area that can be selected or deselected with a checkbox. These blocks are the optional blocks.

   – A free combination option is still available for the optional blocks.

5. Click in the "Control module type variants" area in the "Click here to add a variant" line.
6. Make an entry in the "Variant name" field.
7. Create a combination from the optional blocks that has not yet been used by selecting or deselecting the checkboxes in the line.
8. Click an area outside the line with your mouse.
9. Click the "Generate" button.

Deleting variants

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Create templates" command.

   The "Create templates" tab is opened.

2. Select the "Variants" tab.
3. Select an entry in the "Control module types" column.
4. Click on the interface with a cross in a line in the "Control module type variants" area.
5. Click the "Generate" button.

Result

The variants of the control module types are created as follows:

- The "Automation Tree" tab

  "Automation Tree > Project > Comos_Lib Project library > VAR Variants"

- "Units" tab

  "@Template Copy templates > @PCS 7 Control module library > Comos_Lib Library folder"

See also

Creating templates of individual control unit types (Page 43)

Engineering in COMOS (Page 46)

3.9 Engineering in COMOS

3.9.1 Hardware engineering in the Navigator

3.9.1.1 Creating a SIMATIC station and equipping it with hardware

Requirement

- The templates of the control module types have been created.

  See section Creating templates of individual control unit types (Page 43).
- The "Units" tab is visible in the Navigator.
- A factory/building/production EN standard has been created.

Creating a SIMATIC station

1. In the Navigator, select the "0 Factory/building/production EN standard > L001 > G001 > R001 Room, control center" object.
2. Select the "New > Cabinet" command from the context menu of the room.

   The cabinet is created below the room.
3. In the context menu of the cabinet, select the command "New > SPS > S7-400 station S7-400".

   The station is created below the cabinet.

Equipping the SIMATIC station with hardware

1. Select the station in the Navigator.
2. Select the "New > ..." command in the context menu of the station.
   - Select a rack. The rack is used to mount the assemblies.
3. Select the "New > Power supplies > ..." command in the context menu of the rack.
   - Select a power supply.

4. Select the "New > Central processing units > ..." command in the context menu of the rack.
   - Select a central processing unit (CPU).

     Each central processing unit is equipped with at least one submodule. The submodule is created automatically in addition.

     See also section SIMATIC station: Configuring the central processing unit (CPU) (Page 47).
   - Depending on the central processing unit: Create further modules below the CPU, if required; for example, an interface module for a bus system.
   - Optional: Create the "Nodes" under the bus system. In the example according to this documentation, the bus is located here: "Locations" tab, "Site > Building > Floor > Room > Cabinet > Station > Rack > CPU > Submodule > Bus".
5. Select the "New > ... command in the context menu of the rack.
   - Select an input module from one of the following folders: "Digital modules", "Analog modules", "Function modules".
   - Select an appropriate output module for the input module from one of the following folders: "Digital modules", "Analog modules", "Function modules".
6. If necessary, assign the rack to a location where it should be assembled.

See also

Distributed I/O: Equipping interface modules with I/O modules (Page 49)

3.9.1.2 SIMATIC station: Configuring the central processing unit (CPU)

Requirement

- A SIMATIC station is created and equipped with hardware.

  See chapter Creating a SIMATIC station and equipping it with hardware (Page 46).
- In the rack, at least one central processing unit (CPU) is created.
- The "Units" tab is visible in the Navigator.

Procedure

1. Select the central processing unit (CPU) in the Navigator.
2. Open the properties of the central processing unit (CPU).
3. In the "Name" field, change the value in accordance with the slot number of the CPU in the rack. The value "3" is entered for most configurations.

   The value depends on the number of slots occupied by the power supply. There are some power supplies which occupy only one slot. In this case, the CPU receives the value "2" in the "Name" field.

   The decisive factor is whether the slots are occupied consecutively in the rack. A rack with an empty slot in the middle cannot be saved in SIMATIC HWKonfig.

See also

Information on the identification of SIMATIC components (Page 29)

3.9.1.3 Creating an interface module as distributed I/O (DP slave)

Requirement

- A SIMATIC station with a CPU and bus system has been created.

  See section Creating a SIMATIC station and equipping it with hardware (Page 46).
- The "Locations" tab is displayed in the Navigator.

Creating a distributed I/O (DP slave) using the base data

To create the distributed I/O in the project, proceed as follows:

1. Select the engineering object "0 Factory/building/production EN standard".
2. Select the "New > Field location" command from the context menu of the object.

   The field location is created below the object.
3. Select the "New > Field location, Sensors" command from the context menu of the field location.

   The sensors field location is created below the field location.
4. Select the "New > General > New object" command in the context menu of the sensors field location.
5. Select the "General" tab in the properties of the new object.

6. Set a reference in the "Base object" field.

   To do this, proceed as follows:

   – Switch to the "Base objects" tab in the Navigator.
   – Open the node "@01 Material > EIC > 03 > SIEMENS AG > A > 02 > 01 > 01 > 02 Distributed I/O SIMATIC ET200".
   – Select an interface module; for example, in "02 Distributed I/O SIMATIC ET200 >04 > 01 Interface modules > ...".
   – Drag the interface module into the "Base object" field of the "General" tab that you have opened.

7. Confirm your entries.

See also

Assigning a field bus to a DP slave using a task (Page 53)

Distributed I/O: Equipping interface modules with I/O modules (Page 49)

3.9.1.4 Distributed I/O: Equipping interface modules with I/O modules

Requirement

- The interface modules have been created as distributed I/O.

  See section Creating an interface module as distributed I/O (DP slave) (Page 48).
- The "Units" tab is visible in the Navigator.

Procedure

1. In the Navigator, select the interface module of the distributed I/O.

   According to the examples in this documentation, the interface module is available as follows:

   "0 Factory/building/production EN standard > O001 Field location > B001 Field location, Sensors > (Interface module)".

2. Select the "New > ... command in the context menu of the interface module.
   – Select an input module.

     Select a module marked with "E" for it.
   – Select an output module.

     Then select a module marked with "A" that is suitable for the input module.

See also

Distributed I/O: Configuring IO modules (Page 50)

3.9.1.5 Distributed I/O: Configuring IO modules

Requirement

- The central processing unit (CPU) is configured

  See chapter SIMATIC station: Configuring the central processing unit (CPU) (Page 47).
- The interface modules of the distributed I/O are equipped with IO modules.

  See chapter Distributed I/O: Equipping interface modules with I/O modules (Page 49).
- The "Locations" tab is visible.

Procedure

1. In the Navigator, select the module below the interface module.
2. Open the properties of the module.
3. In the "Name" field, change the value in accordance with the slot number in the interface module. A value of "4" or higher is entered for most configurations. Example:
   - The input module receives the name "4".
   - The output module receives the name "5".

   This has the result that the names of the signal modules correspond to the respective slot number in the interface module.

See also

Information on the identification of SIMATIC components (Page 29)

3.9.1.6 Creating the SIMATIC station with redundant hardware

Requirement

- The control module types are managed.

  See chapter Management of individual control unit types in COMOS (Page 39).
- The "Units" tab is visible in the Navigator.
- A factory/building/production EN standard has been created.

Creating the SIMATIC station that is prepared for redundant hardware.

1. In the Navigator, select the "0 Factory/building/production EN standard > L001 > G001 > R001 Room, control center" object.
2. Select the "New > Cabinet" command from the context menu of the room.

   The cabinet is created below the room.

3. Select a station in the context menu of the cabinet that has been prepared for redundant hardware.

   Example: "New > SPS > S7-400 station S7-400H".

   The station is created below the cabinet.

4. Optional: If the station is not available in the context menu of the station, use the following base object:

   "@03 Structures > BAS > CH01 > S > 02 > S7-400H SIMATIC S7-400H".

Equipping the SIMATIC station with redundant hardware

1. Select the station in the Navigator.
2. Select the "New > ..." command in the context menu of the station.
   - Select a rack.

     Example: Select a rack of the UR2-H type:

     "6ES7 400-2JA10-0AA0 rack, UR2-H, alu"
3. Select the "New > Power supplies > ..." command in the context menu of the rack.
   - Select a power supply.
4. Select the "New > Central processing units > ..." command in the context menu of the rack.
   - Select a central processing unit (CPU).

     Example: Select a CPU of the 417H type:

     "6ES7 417-4HL04-0AB0 CPU 417H"

     Two "Synchronization module" elements are prepared under the CPU 417H. The two synchronization modules are created by the user. The "Name" field of the synchronization modules contains the number of the slot occupied by the synchronization module in the CPU.
   - Optional: Create the "Nodes" under the bus system. In the example according to this documentation, the bus is located here: "Locations" tab, "Site > Building > Floor > Room > Cabinet > Station > Rack > CPU > Submodule > Bus".
5. Select the "New > ... command in the context menu of the rack.
   - Select an input module.

     For this, select a module marked with "E" from one of the following folders: "Digital modules", "Analog modules", "Function modules".
   - Select an output module.

     Then select a module marked with "A" that is suitable for the input module from one of the following folders: "Digital modules", "Analog modules", "Function modules".
6. If necessary, assign the rack to a location where it should be assembled.
7. Create a second rack. Use the same procedure as described for the first rack.

See also

Creating interface modules for redundant hardware (Page 52)

3.9.1.7 Creating interface modules for redundant hardware

Requirement

- A SIMATIC station for redundant hardware has been created, for example, an S7-400H.

  See section Creating the SIMATIC station with redundant hardware (Page 50).
- The "Locations" tab is displayed in the Navigator.

Creating a distributed I/O (DP slave) using the base data

To create the distributed I/O in the project, proceed as follows:

1. Select the engineering object "0 Factory/building/production EN standard".
2. Select the "New > Field location" command from the context menu of the object.

   The field location is created below the object.
3. Select the "New > Field location, Sensors" command from the context menu of the field location.

   The sensors field location is created below the field location.
4. Select the "New > General > New object" command in the context menu of the sensors field location.
5. Select the "General" tab in the properties of the new object.
6. Set a reference in the "Base object" field.

   To do this, proceed as follows:
   - Switch to the "Base objects" tab in the Navigator.
   - Open the node "@01 Material > EIC > 03 > SIEMENS AG > A > 02 > 01 > 01 > 02 > 04 ET 200M > 01 Interface modules"
   - Select an interface module of the "IM 153-2 HF" type.
   - Drag the interface module into the "Base object" field of the "General" tab that you have opened.
7. Confirm your entries.
8. Create a second interface module of the "IM 153-2 HF" type.
9. Confirm your entries. Use the same procedure as described for the first interface module.

See also

Assigning redundant fieldbus using a task (Page 54)

3.9.2 Editing hardware engineering using engineering tasks

3.9.2.1 Assigning a field bus to a DP slave using a task

Requirement

- The hardware engineering has been performed in the Navigator.

  Hardware engineering in the Navigator (Page 46)
  - A SIMATIC station is created and equipped with hardware.
  - An interface module of the distributed I/O is equipped with I/O modules.
- The "Locations" tab is displayed in the Navigator.

Assigning a bus

To assign the interface module of the distributed I/O to a bus, proceed as follows:

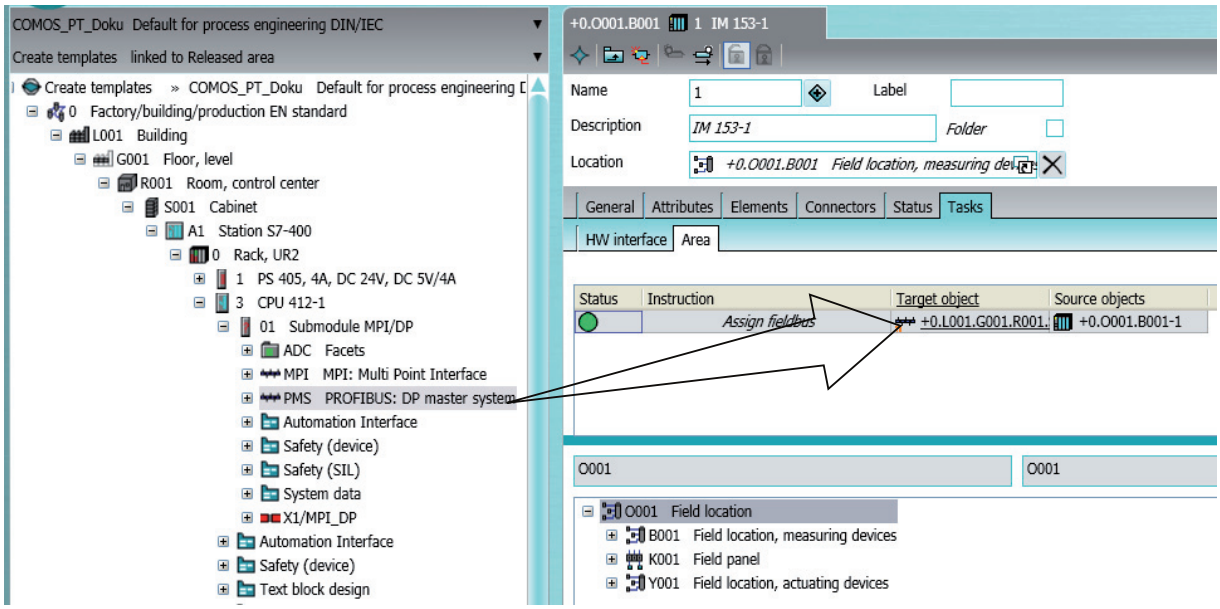
1. Select the interface module in the Navigator.

   The interface module is located here in the example: "Locations" tab, "Site > Building > Field location > Sensors/Actuators > Interface module".
2. Open the properties of the interface module.
3. Open the "Tasks > Areas" tab in the properties.
4. Check to see if the engineering task "Assign fieldbus" is available.

   Create the engineering task "Assign fieldbus" if it is not available.
5. Select the SIMATIC station in the Navigator.
6. Open the structure below the SIMATIC station until the bus is visible.

   The structure is determined in the following step: Auto hotspot

7. Drag the "PROFIBUS" object or a similar object to the "Target object" column:



8. Click the "Execute" button in the "Tasks > Areas" tab.

See also

Software engineering in the Navigator (Page 56)

3.9.2.2 Assigning redundant fieldbus using a task

Requirement

- The hardware engineering has been performed in the Navigator.

  See section Hardware engineering in the Navigator (Page 46).

  – A SIMATIC station is created for redundant hardware and equipped with hardware.

  – The redundant interface modules of the distributed I/O are equipped with I/O modules.
- The "Locations" tab is displayed in the Navigator.

Objective

The first part system of the S7-400H is connected to one of the two PROFIBUS DP interfaces with the engineering task "Assign fieldbus". Then the second part system of the S7-400H is connected to the second PROFIBUS DP interface with the engineering task "Assign redundant fieldbus".

Assigning first bus and redundant bus

1. Select the first interface module in the Navigator.

   In the example according to this documentation, the interface module is located here: "Locations" tab, "Site > Building > Field location > Sensors/Actuators > Interface module".
2. Open the properties of the interface module.
3. Open the "Tasks > Areas" tab in the properties.
4. Check to see if the engineering task "Assign fieldbus" is available.

   Create the engineering task "Assign fieldbus" using the context menu if it is not available.
5. Check to see if the engineering task "Assign redundant fieldbus" is available.

   Create the engineering task "Assign redundant fieldbus" using the context menu if it is not available.
6. Select the SIMATIC station in the Navigator.
7. Open the structure below the SIMATIC station until the first bus is visible.
8. Drag the "PROFIBUS" object or a similar object to the "Assign fieldbus" line in the "Target object" column.
9. Open the structure below the SIMATIC station until the redundant bus is visible.
10. Drag the "PROFIBUS" object or a similar object to the "Assign redundant fieldbus" line in the "Target object" column.
11. Click the "Execute" button in the "Tasks > Areas" tab.
12. Confirm your entries.

Result

The redundant fieldbus is taken into consideration in the "Generate" step. See section Generating COMOS data in the Automation Tree (Page 77).

See also

Software engineering in the Navigator (Page 56)

3.9.3 Software engineering in the Navigator

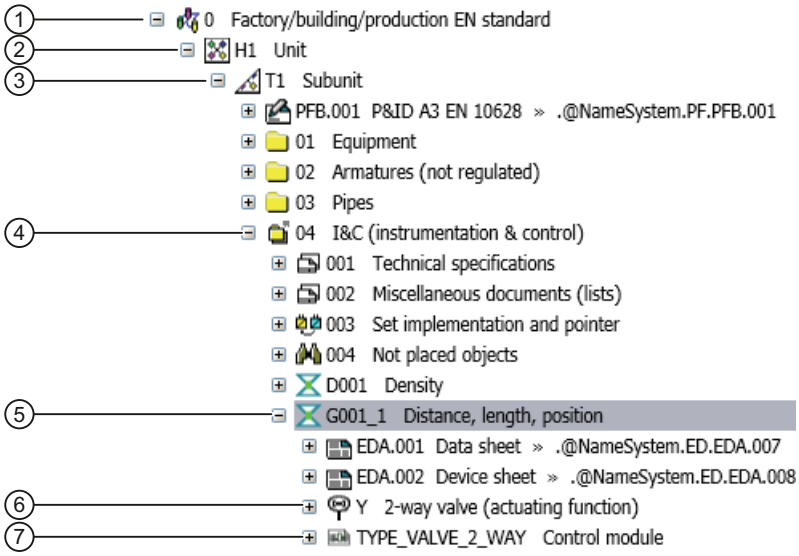
3.9.3.1 Notes on the plant hierarchy

Requirement

- The plant hierarchy from PCS 7 was assigned in COMOS.
  See section Assigning a plant hierarchy in COMOS (Page 36).
- A COMOS engineering project is open.
  See section Initializing a COMOS engineering project (Page 37).

COMOS structure in comparison to plant hierarchy in PCS 7

The plant hierarchy in PCS 7 can include up to 8 levels. A plant hierarchy with four levels is used in this example:



①	Site Part of the plant hierarchy.
②	Unit Part of the plant hierarchy.
③	Part unit Part of the plant hierarchy.
④	Folder COMOS-specific structure element.
⑤	Position Part of the plant hierarchy.

⑥	COMOS process tag (function)
⑦	PCS 7 control module (CM).

See also

Creating and configuring COMOS process tags (Page 57)

3.9.3.2 Creating and configuring COMOS process tags

Requirement

- The plant hierarchy is clear.

  See section Notes on the plant hierarchy (Page 56).
- The "Units" tab is visible.

Creating and configuring process tags

See the "EI&C Operation" manual, keyword "Basic Engineering", on the topic of creating and configuring process tags.

Specifically, the following steps are necessary:

- Select the folder "04 I&C (Instrumentation & Control)".
- Create a position.
- Create COMOS process tag (function).

  Do not use a function template.
- Select the "02 Signal engineering" folder below the function.
- Create signals.
- Create the channel request "N standard channel request for control" for each signal.

  This channel request is used to link the signal to the DP Slave.

See also

Linking hardware and software: Implementing channels (Page 60)

Creating a control module (Page 58)

3.9.3.3 Creating a control module

Requirement

- The COMOS process tag has been created and configured.

  See section Creating and configuring COMOS process tags (Page 57).

Procedure

Control modules are created using engineering tasks. See section Editing software engineering using engineering tasks (Page 61).

3.9.3.4 Creating and using messages

Requirement

- A control module has been created using an engineering task.

  See chapter Editing software engineering using engineering tasks (Page 61).
- The "Units" tab is opened.

Definition

The term 'messages' is an abbreviation for block messages. The following only refers to messages. See chapter Structure of control module types (Page 39).

If the control module type contains a correspondingly prepared message, the message is also created automatically when the control module is created.

Exception: No mapping codes are available.

Creating and editing a message

If you wish to create an additional message for a control module, proceed as follows:

1. Select a control module.
2. Select the "New > 02 Message" command in the context menu.
3. Optional: Select the block below the control module and then the command "New > 02 Message" in the context menu.
4. Open the properties of the message.
5. Select the "Attributes > PCS 7" tab.
6. Edit the fields.
7. Confirm your entries.

Bulk editing of messages

1. Select the "04 I&C (Instrumentation & Control) > 002 Other documents" folder in the part unit.
2. Select the "New > EFP.003 Message list" command in the context menu.
3. Optional: Create the message list by using the following template: "@03 Structures > AES > 300 > 07 Report folder > MSGL Message list".
   - Create a report template in the base object that uses this base object or the report stored there.
   - Create a new document in the engineering project that uses this report template.

     You will find additional information on this topic in the manual "Platform Getting Started", keyword "The "New document" command".
4. Open the message list.
5. Filter and select the entries.
6. Edit the fields.
7. Confirm your entries.

3.9.3.5 Using user-defined attributes

Definition

See section Requirements in PCS 7 (Page 33).

Edit user-defined attribute

1. Open the properties of a control module or a control variable below the control module.
2. Switch to the "Attributes > PCS 7" tab.
3. Edit the user-defined attributes. The following user-defined attributes may be present:
   - Type
   - Name
   - Comment: available in COMOS at "Parameter" object
   - Symbol: available in COMOS at "Block symbol" attribute
   - Cycle time

     The cycle time is not included in the user-defined attributes according to the more restrictive definition. However, the cycle time is managed as a user-defined attribute as part of the import/export.
4. Confirm your entries.

Bulk editing of user-defined attributes

1. Select the "04 I&C (Instrumentation & Control) > 002 Other documents" folder in the part unit.
2. Select the "New > EFP.004 User-defined attribute list" command in the context menu.
3. Optional: Create the message list by using the following template: "@03 Structures > AES > 300 > 07 Report folder > UDAL User-defined attribute list".
   - Create a report template in the base object that uses this base object or the report stored there.
   - Create a new document in the engineering project that uses this report template.

     You will find additional information on this topic in the manual "Platform Getting Started", keyword "The "New document" command".
4. Open the list of user-defined attributes.
5. Filter and select the entries.
6. Edit the fields.
7. Confirm your entries.

3.9.3.6 Linking hardware and software: Implementing channels

Requirement

- The hardware engineering has been performed.

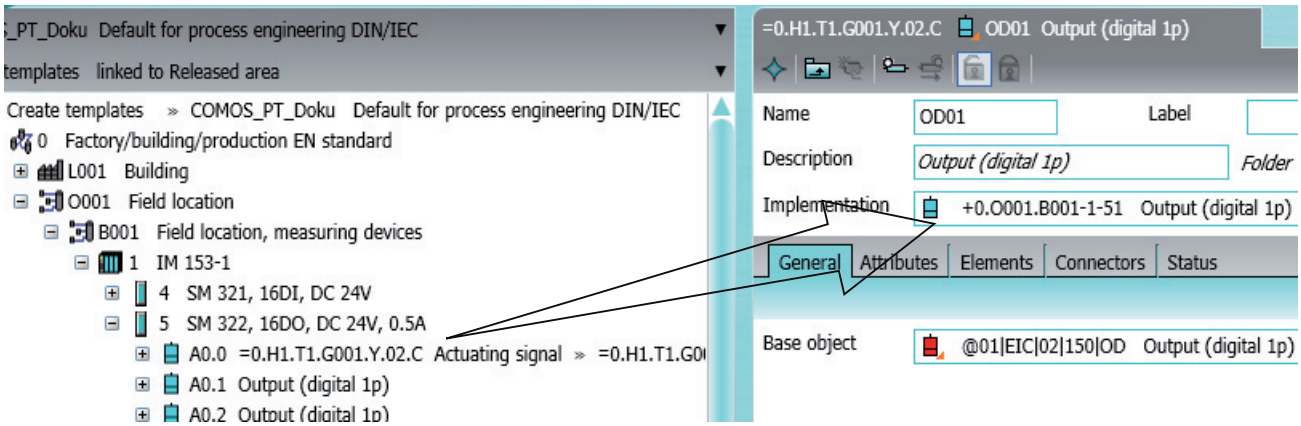
  See section Editing hardware engineering using engineering tasks (Page 53).
- The COMOS process tags have been created and configured.

  See section Creating and configuring COMOS process tags (Page 57).

Implementing the signal using channel assignment

1. Select the signal in the "02 Signal engineering" folder in the "Units" tab.
2. Open the properties of the signal.
3. Switch to the "Locations" tab.
4. Select an interface module below the distributed I/O in the "Locations" tab.

5. Drag a channel of the interface module in the "Implementation" field to the properties of the signal:



6. Confirm your entries.

See also

Editing software engineering using engineering tasks (Page 61)

3.9.4 Editing software engineering using engineering tasks

3.9.4.1 Objective

Requirement

- The software engineering has been performed in the Navigator.

  See section Software engineering in the Navigator (Page 56).

Objective

With the help of an engineering task, a control module is generated from a template below a position in the following way.

See also

Show "SW Interface" tab and create tasks (Page 62)

3.9.4.2 Show "SW Interface" tab and create tasks

Requirement

- The templates of the control module types have been created.

  See section Creating templates of individual control unit types (Page 43).
- The software engineering has been performed.

  Software engineering in the Navigator (Page 56)
- The "Units" tab is visible.

Show the "SW Interface" tab

1. Select the "04 I&C (Instrumentation & Control)" folder in the "Units" tab. The folder is located in the following place, as per the example:

   "0 Factory/building/production EN standard > H1 Unit > T1 Part unit".
2. Select a position in the "04 I&C (Instrumentation & Control)" folder.
3. Open the "Tasks" tab in the properties of the position.
4. Create or select the "SW Interface" tab as follows:
   - If the "SW Interface" tab is not available:

     In the "Tasks" tab, open the context menu on any tab.

     Select "New > Copy and paste control module" in the context menu.

     The "SW Interface" tab opens.
   - If the "SW Interface" tab is available:

     Select the "SW Interface" tab.

Checking and creating engineering tasks

1. Select the "SW Interface" tab.
2. Check whether the following engineering tasks are available:
   - "Copy and paste control module"
   - "Assigned PLC"
3. Create missing tasks using the "New" context menu.

See also

Assigning a control module type to a position (Page 63)

Assigning a control module type to a function (Page 64)

Assigning a station to a position (Page 65)

3.9.4.3 Assigning a control module type to a position

Requirement

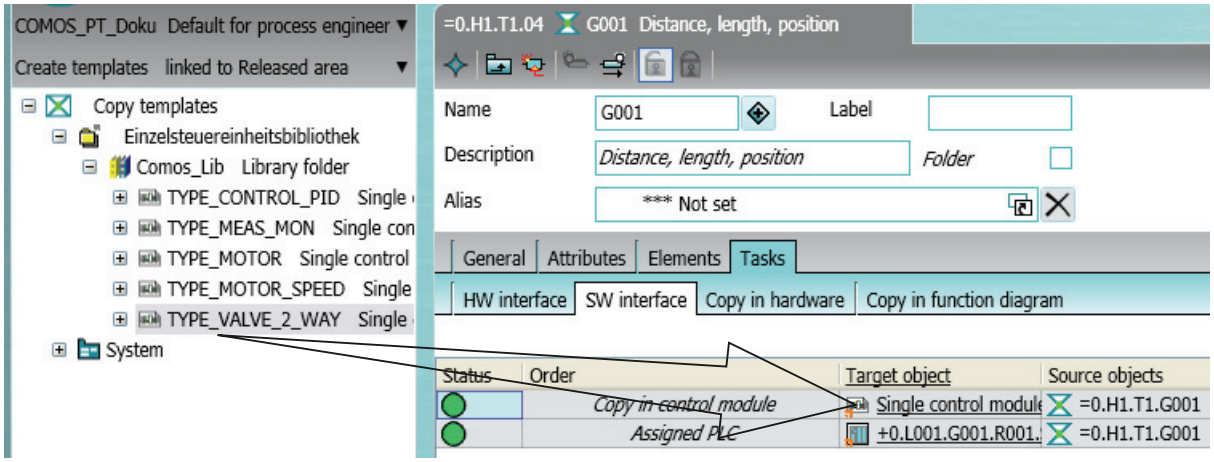
- The hardware engineering has been performed.

  See section Editing hardware engineering using engineering tasks (Page 53).
- The "SW Interface" tab of a position is visible and the engineering tasks have been created.

  See section Show "SW Interface" tab and create tasks (Page 62).
- The "Units" tab is visible.

Configuring and executing an engineering task

1. Keep the "SW Interface" tab of the position open.
2. Switch to the "Templates" tab in the Navigator.
3. Select the "@Template Control module library > Comos_Lib library folder" object.
4. Select the template of a control module in the "Comos_Lib Library folder" folder.
5. Drag&drop the template in the "Copy and paste control module" line in the "Target object" field.



6. Click the arrow of the "Execute" button in the upper, right task area.

   The button has the following icon: ✓ ▾
7. Select the "Execute (selected)" entry in the "Execute" list.
8. Confirm your entries.

   Executed tasks are marked with a green signal light in the "Status" column.

Result

- The copied and pasted control module is displayed below the position in the Navigator structure. See also the overview in Notes on the plant hierarchy (Page 56).
- A mapping table is below the copied and pasted control module. The mapping table is edited by the administrator. See also section Configuring the mapping table of the signal designations (Page 72).

See also

Assigning a control module type to a function (Page 64)

Assigning a station to a position (Page 65)

3.9.4.4 Assigning a control module type to a function

Requirement

- The hardware engineering has been performed.

  See section Editing hardware engineering using engineering tasks (Page 53).
- The "SW Interface" tab of a position is visible and the engineering tasks have been created.

  See section Show "SW Interface" tab and create tasks (Page 62).
- The "Units" tab is visible.

Configuring and executing an engineering task

1. Keep the "SW Interface" tab of the function open.
2. Proceed as described here.

   Assigning a control module type to a position (Page 63)
3. Confirm your entries.

   Executed tasks are marked with a green signal light in the "Status" column.

Result

- The copied and pasted control module is displayed below the function in the Navigator structure. See also the overview in Notes on the plant hierarchy (Page 56).
- A mapping table is below the copied and pasted control module. The mapping table is edited by the administrator. See also section Configuring the mapping table of the signal designations (Page 72).

See also

Assigning a station to a position (Page 65)

3.9.4.5 Assigning a station to a position

Requirement

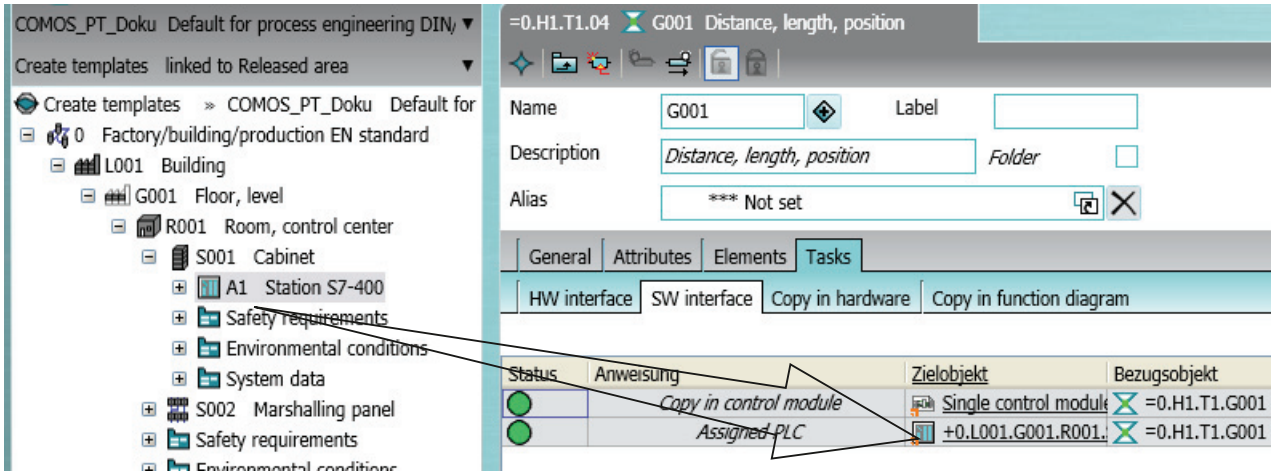
- The "SW Interface" tab of a position is visible and the engineering tasks have been created.

  See section Show "SW Interface" tab and create tasks (Page 62).
- The hardware engineering has been performed.

  See section Editing hardware engineering using engineering tasks (Page 53).
- The "Units" tab is visible.

Configuring and executing an engineering task

1. Keep the "SW Interface" tab of the position open.
2. Switch to the "Locations" tab in the Navigator.
3. Select the "0 Factory/building/production EN standard > L001 Building > G001 Floor, story > R001 Room, control center > S001 Cabinet > Station" object.
4. Drag&drop the station in the "Assigned PLC" line in the "Target object" field.



5. Click the arrow of the "Execute" button in the upper, right task area.

   The button has the following icon: ✓ ▾
6. Select the "Execute (selected)" entry in the "Execute" list.
7. Confirm your entries.

   Executed tasks are marked with a green signal light in the "Status" column.

See also

Assigning a control module type to positions in bulk (Page 66)

3.9.4.6 Assigning a control module type to positions in bulk

Requirement

- The hardware engineering has been performed.

  See section Editing hardware engineering using engineering tasks (Page 53).
- The "Units" tab is visible.

Bulk processing of engineering tasks

1. Select a part unit.
2. Open the "Tasks" tab in the properties of the part unit.
3. Make sure that the "SW Interface" tab is visible and the engineering tasks have been created.

   See section Show "SW Interface" tab and create tasks (Page 62).
4. Select the "SW Interface" tab and proceed as described here:

   Assigning a control module type to a position (Page 63)

   The engineering task includes all positions below the part unit.

3.9.5 Using symbol tables

3.9.5.1 Creating a symbol table

Requirement

- The software engineering has been performed.

  See section Hardware engineering in the Navigator (Page 46).
- The software engineering has been performed.

  See section Editing software engineering using engineering tasks (Page 61).
- The signals of the positions have been edited.

  The symbolic names do not need to be edited: symbolic addresses specified by the plant hierarchy are used.

  See the "EI&C Operation" manual, keyword "Basic Engineering", on the topic of creating and configuring positions.

Creating a symbol table

To add a symbol table to a station, proceed as follows:

1. Switch to the "Locations" tab in the Navigator.
2. In the "Station" context menu of the Navigator, select the command "New > ST Symbol table".

Result

- The symbol table is created below the "Station" object.
- The signals are linked to the symbol table, but are not yet visible in the table.

See also

Importing symbolic addresses into a symbol table (Page 67)

3.9.5.2 Importing symbolic addresses into a symbol table

Requirement

- A symbol table has been created.

  See section Creating a symbol table (Page 66).

Importing symbolic addresses into a symbol table

You can also import the symbolic addresses from a symbol table stored in the file system into the symbol table:

1. Select the "Symbol table > Import" command in the context menu of the symbol table.
2. In the file system you select the symbol table from which you want to import the symbolic addresses.

See also

Refreshing the symbol table (Page 67)

3.9.5.3 Refreshing the symbol table

Requirement

- A symbol table has been created.

  See section Creating a symbol table (Page 66).

Procedure

To refresh the symbol table, you can choose between the following options:

- To refresh all symbolic addresses in the symbol table, select the command "Symbol table > Refresh symbol table" from the context menu of the symbol table.
- To refresh the symbolic addresses of individual positions in the symbol table, select these positions and then select the "Symbol table > Refresh symbol table" command from the context menu of the object.

Result

The symbolic addresses are written into the symbol table.

3.9.6 Engineering in the function diagram

3.9.6.1 Creating and opening a function diagram

Requirement

- The hardware engineering has been performed.

  See section Hardware engineering in the Navigator (Page 46).

  See section Editing hardware engineering using engineering tasks (Page 53).
- The software engineering has been performed.

  See section Software engineering in the Navigator (Page 56).

  See section Editing software engineering using engineering tasks (Page 61).
- The "Units" tab is visible.

Procedure

1. Select a position.

   For the default structure of the "Units" tab, see section Notes on the plant hierarchy (Page 56).
2. Select the "New > EFF.001 Function diagram IEC, A3" command from the context menu of the position.
3. Open the function diagram.

See also

Editing control modules on the function diagram (Page 69)

3.9.6.2 Editing control modules on the function diagram

Requirement

- The function diagram has been created and is open.

  See chapter Creating and opening a function diagram (Page 68).

Procedure

1. Select the position below which the function is located.
2. Select a control module below the position.
3. Drag&drop the control module to the function diagram.

   Note: The COMOS object "Control module" can be placed in different ways, similar to the COMOS object "Process tag". If you drag the control module to the function diagram, a block is created on the function diagram.
4. Place generic blocks on the function diagram.

   You can find more information on this topic in the "Logical" manual, keyword "Connecting blocks".

   – The generic blocks OR, AND and XOR are available in the menu of the function diagram. Only these generic blocks may be used.
5. Select a COMOS process tag below the position.
6. Select the "02 Signal engineering" folder below the COMOS process tag.
7. Select a signal in the "02 Signal engineering" folder.
8. Drag the signal onto the function diagram.

   The signal has been implemented on the channels in a previous step. See chapter Linking hardware and software: Implementing channels (Page 60).

   If the data type of the signal does not match the data type of the block connection, a connection cannot be established. You can find additional information on this topic in the "Logical" manual, keyword "Connection types with block connections" and keyword "Permitted connector type combinations".
9. Connect the objects on the function diagram.

NOTICE
Connecting signals directly with the block You cannot connect a hardware signal initially with a generic block and then connect the generic block with the block. Instead you can only connect the hardware signal directly with the input of the placed block. Only this procedure is exported during the export to PCS 7.

Result

The interconnected function diagrams can be recorded in the "Generate" step. Only the export from COMOS to PCS 7 is supported. See chapter Generating COMOS data in the Automation Tree (Page 77).

See also

Editing interlock logic on function diagrams (Page 70)

3.9.6.3 Editing interlock logic on function diagrams

Requirement

- A control module has been edited on a function diagram.

  Editing control modules on the function diagram (Page 69)
- The hardware driver block and its connections must be taken into consideration in the plant connection.

Overview of interlock logic in PCS 7

Interlock logic refers to control modules and their blocks that are interlocked. Example:

- The VALVE_1 control module controls a valve.
- The PUMP_1 control module controls a pump.
- Dependency: If the valve is closed, the pump must be switched off.

This structure is implemented as follows in PCS 7:

- There is a function diagram below the VALVE_1 control module which maps the blocks of VALVE_1.
- There is a function diagram below the PUMP_1 control module which maps the blocks of PUMP_1.
- The blocks of VALVE_1 and PUMP_1 are connected by means of parameter value.

This structure is recreated in COMOS in the example according to this documentation.

Creating and connecting control modules

1. Create a function diagram for the first control module, for example, for VALVE_1.

   See chapter Editing control modules on the function diagram (Page 69).
2. Create a function diagram for the first control module, for example, for PUMP_1.

   See chapter Editing control modules on the function diagram (Page 69).
3. Create a generic block on the function diagram for PUMP_1, if necessary. This block is used to connect the links to other blocks.

   It is not permitted to connect hardware signals to the generic block.

4. Create a semi-open connection at the output of VALVE_1.
5. Create a semi-open connection at the input of the generic block on the function diagram for PUMP_1.
6. Select the command "Connection > Memorize" in the context menu at the semi-open connection on the function diagram for VALVE_1.
7. Select the command "Connection > Connect with" in the context menu at the semi-open connection at the input of the generic block on the function diagram for PUMP_1.

See also

Example for interlock logic on function diagrams (Page 71)

Generating COMOS data in the Automation Tree (Page 77)

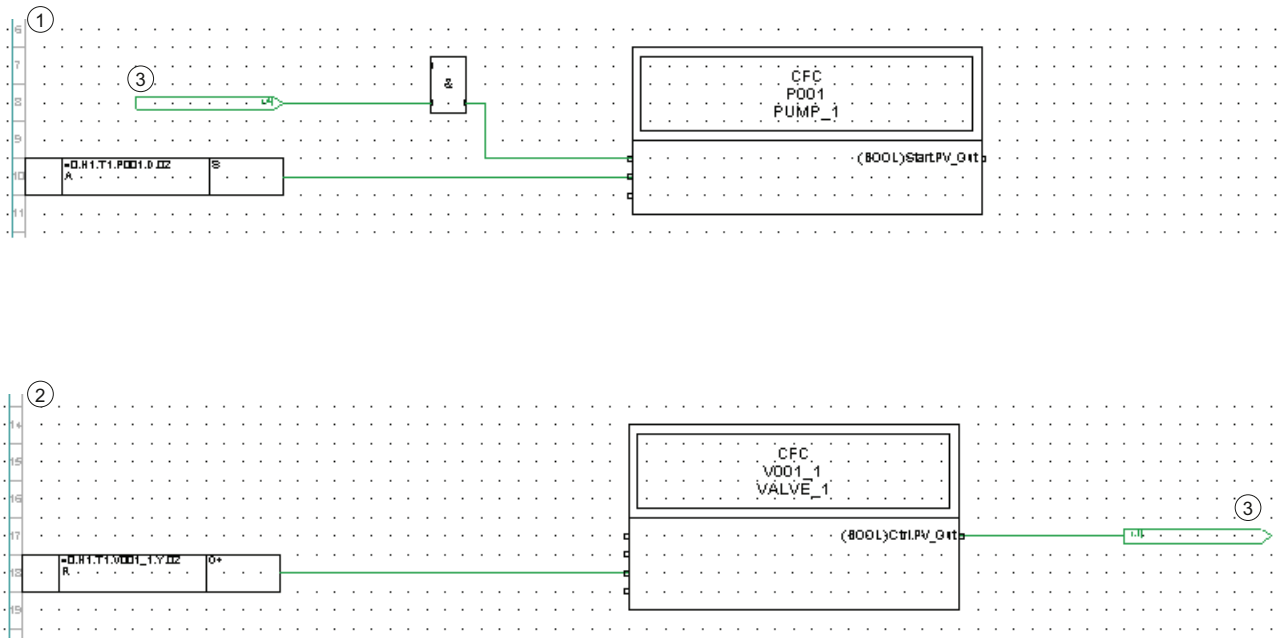
3.9.6.4 Example for interlock logic on function diagrams

Requirement

- An interlock logic has been created on function diagrams.

  See section Editing interlock logic on function diagrams (Page 70).

Example



①	Function diagram of the pump
②	Function diagram of the valve
③	Semi-open connection with link across documents

3.10 Linking attributes from COMOS and PCS 7

3.10.1 Configuring the mapping table of the signal designations

Requirement

- The management of control modules types has been performed.
  See section Management of individual control unit types in COMOS (Page 39).
  – In particular: The templates of the control module types have been created.
- A control module has been copied and pasted.
  See section Creating a control module (Page 58).
- The "Units" tab is visible.

Objective

The mapping table must initially be edited in the engineering data. In so doing, the mapping table is edited on a process tag, for example. The mapping table is subsequently saved in the templates and is then available for all process tags of this type.

Calling a mapping table

1. Switch to the "Units" tab in the Navigator.
2. In the unit structure, navigate to the copied and pasted control module below a position.
3. In the properties of the control module, select the "Attributes > COMOS system" tab.
4. Click the "Open mapping table" button.

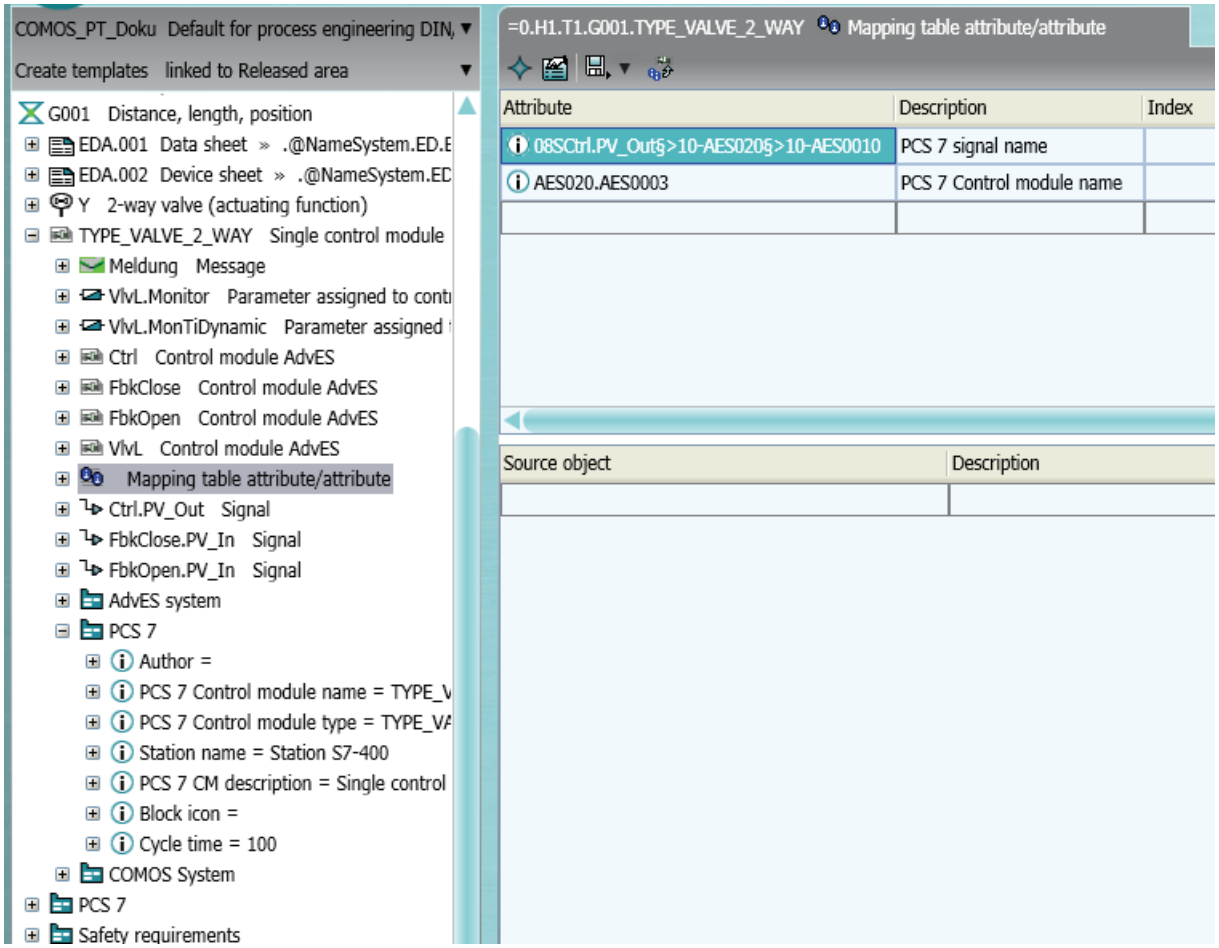
Step 1: Collecting PCS 7 attributes

1. Keep the mapping table open.
2. Open the structure below the control module used in the previous step.

   Only attributes are allowed that are located below the control module, and for which the mapping table was opened. Use the following attributes as per the example in this documentation:

   – Control module, "PCS 7" tab: "PCS 7 control module name"
   – Signal under the control module, "PCS 7" tab: "PCS 7 signal name"

3. Drag&drop this attribute from the Navigator into the "Attribute" column.



4. For each attribute used, double-click in the "Link type" column and select the "Full dynamic" list entry.

   The "Full dynamic" option ensures that all properties of the linked attribute are matched.

Step 2: Assigning COMOS attributes

1. Keep the mapping table open.
2. Select a COMOS process tag below the same position as in the previous step.

   An example of an assignment is available in the section Example of an assignment (Page 76).
3. Drag&drop this attribute from the Navigator into the "Source object" column.
4. Double-click in the "Property" column and select "DisplayValue" to display the value.
5. Check the transferred property in the "Value" column.
6. To store the assignments, click the "Apply" button.

Optional: Assigning COMOS attributes using the navigation assistant

Alternatively, you can select the source object in the navigation assistant.

See section Using the navigation assistant (Page 75).

See also

Adopting a mapping table as a template (Page 76)

3.10.2 Using the navigation assistant

Requirement

- The mapping table of the signal designations is open.

  See section Configuring the mapping table of the signal designations (Page 72).

Objective

The navigation assistant provides a graphical user interface for specifying the source object in the mapping table. It is used if the position of the source object in the unit tree is unknown.

Call

To call the navigation assistant, proceed as follows:

1. Open the properties of a control module.
2. Switch to the "Attributes > COMOS system" tab and click on the "Open mapping table" button.
3. In the mapping table, double-click in the required cell of the "Source object" column.

   The navigation assistant opens. The currently selected control module is entered as the start object.
4. To define the target object step-by-step, proceed as follows:
   - Double-click on an entry in the "Object" column in the right table.

     The selected object is transferred to the "Step" column of the left table.
   - To create another step below the last step, mark the last step in the left table and double-click the required object in the right table.
   - To remove a step from the left table, select the "Remove row" command in the context menu of the row to be deleted.
5. Confirm your entries.

You can find additional information on this topic in the "Queries" manual, keyword "Calculation type: Navigation library expanded".

See also

Example of an assignment (Page 76)

3.10.3 Example of an assignment

Requirement

- The mapping table of the signal designations is open.

  See section Configuring the mapping table of the signal designations (Page 72).

Procedure

The following assignment applies based on the example in this documentation:

"AES0010 PCS 7 signal name"	COMOS process tag, "02 Signal engineering" folder, properties of the signal: "FUP01 Signal data" tab: "FUP001 Signal designation"
"AES0003 PCS 7 control module name"	item: "AES020 PCS 7" tab: "AES0022 PCS 7 hierarchy name"
"AES0011 Station name"	This attribute is controlled by the script. Manual linking is no longer possible.

See also

Adopting a mapping table as a template (Page 76)

3.10.4 Adopting a mapping table as a template

Requirement

- The mapping table of the signal designations is configured.

  See section Configuring the mapping table of the signal designations (Page 72).
- The "Units" tab is visible.

Objective

The mapping table is saved in the templates and is then available for all control modules of this type.

Calling a mapping table

1. In the unit structure, navigate to the copied and pasted control module below a position.
2. In the properties of the control module, select the "Attributes > COMOS system" tab.
3. Click the "Open mapping table" button.
4. Edit the mapping table according to these instructions:

   Reference to the mapping table (Page 100)

Restoring the mapping table

To restore the assignments to the control module type, proceed as follows:

1. Open the fully assigned mapping table of a control module.
2. To restore the assignments to the template of the employed control module type, click on the "Insert in template" icon in the icon bar at the top.

    .
3. Confirm your entries.

If you want to check the assignments in the template, open the mapping table of the control module type in the "@Template templates > @PCS 7 > Library folder" folder.

3.11 Exporting from COMOS to PCS 7

3.11.1 Generating COMOS data in the Automation Tree

Requirement

- Engineering has been performed in COMOS.

  See section Engineering in COMOS (Page 46).

Objective

Apply the COMOS data to the "Automation Tree" tab.

Procedure

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.
2. Select the "Generate" tab.

3. Drag an object of the configured unit from the "Units" or "Locations" tab to the "Start object" field.

   All stations below the selected start object are displayed in the table.

4. Enable all options for the "Filter" control group.

   See also section "Generate" tab (Page 99).

5. In the "Target" control group, select the target project in the list for all stations listed in the "PCS 7 project" column.

   All projects in the "Automation Tree" tab are offered for selection.

6. Optional: Create a new project.

   See section Initializing a COMOS engineering project (Page 37).

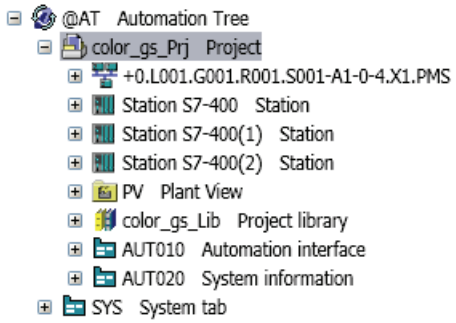
7. Click the "Generate" button to start generation.
   - Errors, warnings, and messages are displayed in the status window after generation.

     See also section Information in the "Data transfer - Generate/import" dialog (Page 94).
   - Depending on the application, the "Unspecified SW hierarchy" folder opens.

Result

- Default structure

  All objects below the selected start object are generated in the "Automation Tree" tab under the specified PCS 7 project.



- Redundant hardware

  The following conditions are checked during generation:

  – If the task "Assign redundant fieldbus" was executed at the ET200M interface module.

  – If the ET200M has the functional classification BC RD.

  If both requirements are met, a second Profibus node is created in the Automation Tree. The required data are retrieved from the target object of the "Assign redundant fieldbus" task. The name of the respective Profibus is written to the following attribute at the Profibus object: "AUT010" tab, "NetRef" attribute. The name of the second Profibus is created with an underscore as prefix, for example, IM 153-2_1.

- Links on function diagrams

  The blocks are created in the following folder: "@AT Automation Tree > Project > PV Plant view". The structure of the "Units" tab is recreated below the folder "PV Plant view".

  The generic blocks added in the function diagram are sorted below the control module.

See also

Exporting a COMOS project (Page 79)

3.11.2 Exporting a COMOS project

Requirement

- The COMOS data have been generated in the Automation Tree.

  See chapter Generating COMOS data in the Automation Tree (Page 77).

Objective

Export the selected objects from the "Automation Tree" tab to PCS 7.

Procedure

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.
2. Select the "Export" tab.
3. In the "Automation Tree project/library" list, select the source project to be exported.
4. Enable all options for the "Filter" control group.
5. Click the "Export" button.

   The "Data transfer - Generate/import" dialog opens. You can find additional information on this topic in the dialog help "Data transfer - Generate/import".
6. Click on "Generate/Synchronize Process Tags from A to B" to generate the control modules.
7. To transfer the plant hierarchy, the hardware and the symbol table, click the "Import objects from A to B" button here.

NOTICE
Export interlock logic to function diagrams The interlock logic on function diagrams is only exported from COMOS to PCS 7. Import from PCS 7 to COMOS is not available.

See also

Exporting the technological hierarchy (Page 81)

Exporting control modules (Page 82)

Exporting the hardware configuration (Page 83)

3.11.3 Exporting in old PCS 7 V7 projects

Working with the "Process tag types" templates from PCS 7 V7

If process tag types are prepared in PCS 7 and the control module types have not yet been used, the following applies:

A new project can be created in COMOS and the process tag types can be imported and used.

In this case, the following applies to importing the COMOS project:

Click on "Generate/Synchronize process tags from A to B" to generate the control modules. If you wish to export cross-chart interconnections, enable the "Make textual interconnections" option in the Import/Export Assistant.

3.11.4 Exporting the technological hierarchy

Objective

Export of the plant hierarchy from the "Automation Tree" tab to PCS 7. If there is already a plant hierarchy in PCS 7, the hierarchy folders will be synchronized.

Procedure

To export the plant hierarchy, proceed as follows:

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.
2. Select the "Export" tab.
3. In the "Automation Tree project/library" list, select the source project to be exported.
4. Enable the "Plant hierarchy" option under "Filter".
5. Click the "Export" button.

   The "Data transfer - Generate/import" dialog opens. You can find additional information on this topic in the dialog help "Data transfer - Generate/import".
6. Click on the "Import objects from A to B" button here.

See also

Effects of exporting the plant hierarchy (Page 81)

3.11.5 Effects of exporting the plant hierarchy

Requirement

- The plant hierarchy has been exported from the "Automation Tree" tab to PCS 7.

  See section Exporting the technological hierarchy (Page 81).

Effects of synchronization on existing hierarchy folders in PCS 7

Hierarchy folders that already exist are not duplicated.

Hierarchy folders that are available in the existing plant hierarchy in PCS 7 but not in the plant hierarchy imported from COMOS result in a query.

- Integrate hierarchy folder: enabled

  The hierarchy folders from COMOS and PCS 7 are consolidated.

- Integrate hierarchy folder: disabled

  The plant hierarchy from COMOS completely overwrites the plant hierarchy in PCS 7 and deletes all additional hierarchy folders of PCS 7. The result is that all control modules in old hierarchy folders are also deleted in the PCS 7 project.

Effects of synchronization on new hierarchy folders in PCS 7

Hierarchy folders that are not available in PCS 7 are added in the plant hierarchy.

3.11.6 Exporting control modules

Requirement

- The plant hierarchy must be exported from COMOS and matched with the plant hierarchy from PCS 7.

Procedure

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.

2. Select the "Export" tab.
3. In the "Automation Tree project/library" list, select the source project to be exported.
4. Enable the "Control modules" option under "Filter".

   The "Plant hierarchy" option is enabled because control modules can only be exported together with the plant hierarchy.

5. Click "Export" to start the export.

   The "Data transfer - Generate/import" dialog opens. You can find additional information on this topic in the dialog help "Data transfer - Generate/import".

6. Click on the "Generate/Synchronize Process Tags from A to B" button here.
7. Confirm your entries.

3.11.7 Exporting the hardware configuration

Requirement

- The hardware must be available in the "Automation Tree" tab.

Procedure

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.
2. Select the "Export" tab.
3. In the "Automation Tree project/library" list, select the source project to be exported.
4. Enable the "Hardware" option under "Filter".
5. Click the "Export" button.

   The "Data transfer - Generate/import" dialog opens. You can find additional information on this topic in the dialog help "Data transfer - Generate/import".
6. Click on the "Import objects from A to B" button here.

Result

The hardware configuration is exported to the PCS 7 project. The "Symbol" entries in the symbol table are completed with the "PCS 7 signal name" values from COMOS.

Note

Do not rename the Profibus

Once you have exported the hardware from COMOS to PCS 7, the Profibus may no longer be renamed.

3.12 Importing from PCS 7 to COMOS

3.12.1 Importing a PCS 7 project

Requirement

- The export from COMOS to PCS 7 has been performed.

  See section Exporting from COMOS to PCS 7 (Page 77).

Objective

The selected objects from PCS 7 are imported into the "Automation Tree" tab.

Procedure

1. In the menu, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.
2. Select the "Import" tab.
3. To select a PCS 7 project or an XML file as the source for the import, click on the "..." button next to the "PCS 7 project/library" field.
4. Enable the required options under "Filter".
5. In the "Automation Tree project/library" list, select the target project to be imported.
6. Click on the "Import" button.

   The "Data transfer - Generate/import" dialog opens. You can find additional information on this topic in the dialog help "Data transfer - Generate/import".
7. Click on the "Import objects from B to A" button here.

   The project data is generated in the Automation Tree.
8. Confirm your entries.

Result

The PCS 7 project is imported to the COMOS "Automation Tree" tab along with the selected objects. Errors, warnings, and messages relating to the import are displayed in the status window. See also section Information in the "Data transfer - Generate/import" dialog (Page 94).

See also

Importing the technological hierarchy (Page 85)

Importing control modules (Page 86)

Importing the hardware configuration (Page 87)

3.12.2 Importing the technological hierarchy

Procedure

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.
2. Select the "Import" tab.
3. To select the source project to be imported, click on the "..." button in the "PCS 7 project/library" line.

   The "Project selection" dialog opens.
4. Enable the "Plant hierarchy" option under "Filter".
5. In the "Automation Tree project/library" list, select the target project to be imported.
6. Click on the "Import" button.

   The "Data transfer - Generate/import" dialog opens.
7. Click on the "Import objects from B to A" button in the "Data transfer - Generate/import" dialog.
8. Confirm your entries.

Result

The plant hierarchy from PCS 7 is imported into the "Automation Tree" tab. If a plant hierarchy already exists in the "Automation Tree" tab, the "Data transfer - Generate/import" dialog synchronizes the hierarchy folders with each another.

See also

Effects of importing the plant hierarchy (Page 85)

3.12.3 Effects of importing the plant hierarchy

Requirement

- The plant hierarchy from PCS 7 has been imported into the "Automation Tree" tab.

  See chapter Importing the technological hierarchy (Page 85).

Effects of synchronization on existing hierarchy folders

- "Delete in target" option: disabled

  No hierarchy folders are deleted. Hierarchy folders that already exist are not duplicated.
- "Delete in target" option: enabled

  This option is required for reengineering. A working layer is created in COMOS, if:

  – The "Delete in target" option is enabled

  – Deleted or renamed objects have to be taken into consideration

  Refer also to chapter Reengineering for deleted control modules (Page 90).

Effects of synchronization on new hierarchy folders in PCS 7

Hierarchy folders from PCS 7 that are not available in COMOS are added to the plant hierarchy of the "PV Plant view" node in the "Automation Tree" tab.

Interaction between the "Units" and "Locations" tabs

The transfer of data from the "Automation Tree" tab in the "Units" tab or "Locations " tab takes place in separate steps.

3.12.4 Importing control modules

Note

Reimport only of control modules in standard procedures

Standard procedures stipulate that control modules can only be imported to COMOS if they were created previously in COMOS and then exported to PCS 7.

Technically, it is also possible to create a new control module instance in PCS 7 and import it from PCS 7 to COMOS if a control module is used which already exists in COMOS.

Requirements

To import control modules into COMOS, the following information must be available for each control module in PCS 7:

- Plant hierarchy
- Control module type
- Parameters and signals used
- The control module type must already exist in COMOS.

Procedure

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.
2. Select the "Import" tab.
3. Select the PCS 7 source project to be imported.
4. Enable the "Control modules" option under "Filter".
5. In the "Automation Tree project/library" list, select the target project to be imported.
6. Click on the "Import" button.

   The "Data transfer - Generate/import" dialog opens. You can find additional information on this topic in the dialog help "Data transfer - Generate/import".
7. Click on the "Import objects from B to A" button in the "Data transfer - Generate/import" dialog.

Result

The control modules are imported into the "Automation Tree" tab of the current COMOS project.

See also

Importing individual control unit types (Page 41)

3.12.5 Importing the hardware configuration

Requirements

- The hardware is available in the PCS 7 project.

Procedure

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.
2. Select the "Import" tab.
3. Select the PCS 7 source project to be imported.
4. Enable the "Hardware" option under "Filter".
5. In the "Automation Tree project/library" list, select the target project to be imported.

6. Click on the "Import" button.

   The "Data transfer - Generate/import" dialog opens. You can find additional information on this topic in the dialog help "Data transfer - Generate/import".

7. Click on the "Import objects from B to A" button in the "Data transfer - Generate/import" dialog.

Result

The hardware is imported into the "Automation Tree" tab of the current COMOS project.

3.12.6 Matching the automation tree and the "Units" tab

Objective

The data buffered in the "Automation Tree" tab is applied to the "Units" tab and the "Locations" tab.

Procedure

To start the matching process, proceed as follows:

1. In the menu bar, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.

2. Select the "Reengineering" tab.
3. In the "Automation Tree project/library" list, select the source project to be matched.
4. In the "Filter" control group, enable the options for the objects for which you wish to perform matching.
5. Click the "Reengineer" button.

Result

Reimported PCS 7 objects that have already been created in the COMOS project are matched and updated automatically.

3.13 Reengineering (Engineering in PCS 7)

3.13.1 Importing a reengineering project

Requirement

- Management of control module types in COMOS has been performed.

  See chapter Management of individual control unit types in COMOS (Page 39).
- Engineering has been performed in COMOS.

  See chapter Engineering in COMOS (Page 46).
- Attributes from COMOS and PCS 7 are linked.

  See chapter Linking attributes from COMOS and PCS 7 (Page 72).

Procedure

1. In the menu, select the "Plugins > Automation > PCS 7 interface > Import/Export PCS 7" command.

   The "Import/Export PCS 7" tab opens.
2. Select the "Import" tab.
3. To select a PCS 7 project or an XML file as the source for the import, click on the "..." button next to the "PCS 7 project/library" field.
4. Enable the required options under "Filter".
5. In the "Automation Tree project/library" list, select the target project to be imported.
6. Click on the "Import" button.

   The "Data transfer - Generate/import" dialog opens.
7. If you want to take deleted control modules into consideration: Select the option "Delete in target" in the "Data transfer - Generate/import" dialog.

   You can find additional information on this topic in the dialog help "Data transfer - Generate/import".

   See also chapter "Reengineering for deleted control modules".
8. Click on the "Import objects from B to A" button here.
9. Confirm your entries.

Result

- A working layer is created in COMOS, if:
  - The "Delete in target" option is enabled
  - Deleted or renamed objects have to be taken into consideration
- The PCS 7 project is imported to the COMOS "Automation Tree" tab along with the selected objects.
- If required: The "Unspecified SW hierarchy" folder is created in the "Units" tab.
- If required: The "Unspecified DV location" folder is created in the "Locations" tab.
- Errors, warnings, and messages relating to the import are displayed in the status window.

See also

Reengineering for deleted control modules (Page 90)
Reengineering for new control modules (Page 91)
Reengineering for revised control modules (Page 91)
Reengineering for hardware (Page 92)
Adapting the plant hierarchy (Page 93)

3.13.2 Reengineering for deleted control modules

Requirement

- The reengineering project was imported into the Automation Tree.

  See section Importing a reengineering project (Page 89).

Procedure

1. Open the working layer that was created by importing the reengineering project.

   The working layer was created, because the "Delete in target" option was selected.
2. Check the results.
   - The control module was automatically deleted during import in the "Automation Tree" tab.
   - The control module was automatically deleted during import in the "Units" tab.
3. Enable the working layer.

   You can find additional information on this topic in the "COMOS Administration" manual, keyword "Managing working layers".

3.13.3 Reengineering for new control modules

Requirement

- The reengineering project was imported into the Automation Tree.

  See section Importing a reengineering project (Page 89).

Procedure

1. Select the "Automation Tree" tab.
2. Check the results.
   - The control module was automatically created during import in the "Automation Tree" tab.
   - The control module was automatically created during import in the "Units" tab in the "Unspecified SW hierarchy" folder. The PCS 7 structure was recreated in this step.
3. Move the control modules in the "Units" tab from the "Unspecified SW hierarchy" folder to the unit structure. The correct location is indicated by the PCS 7 structure.

3.13.4 Reengineering for revised control modules

Requirement

- The reengineering project was imported into the Automation Tree.

  See section Importing a reengineering project (Page 89).

Procedure

The following procedure applies to:

- Renaming control modules
- Moving control modules

1. Select the "Automation Tree" tab.
2. Check the results.
   - The existing control module was automatically deleted in all COMOS tabs during import.
   - The new control module was automatically created during import in the "Automation Tree" tab in the "Unspecified SW hierarchy" folder.
   - The control module was automatically created during import in the "Units" tab in the "Unspecified SW hierarchy" folder. The PCS 7 structure was recreated in this step.
3. Move the control modules in the "Units" tab from the "Unspecified SW hierarchy" folder to the unit structure. The correct location is indicated by the PCS 7 structure.

3.13.5 Reengineering for hardware

Requirement

- The reengineering project was imported into the Automation Tree.

  Importing a reengineering project (Page 89)

Procedure

A hardware component is considered new in reengineering when the following applies:

- No COMOS object is selected in the "AUT020 System Information" tab in the "AUT0001 Comos object" attribute.

1. Select the "Automation Tree" tab.
2. Check the results.

   All hardware components were automatically created during import in the "Automation Tree" tab. In addition, the following applies:

   For stations and components of a station:

   - In addition to the "AUT0001 Comos object" attribute, the "Item number" attribute was automatically checked during import. If an MLFB that exists in COMOS was found in the "Item number" attribute, the PCS 7 object is assigned to the COMOS object.
   - All other hardware components were automatically created during import in the "Unspecified DV location" folder in the "Locations" tab.

   The following applies for bus systems and nodes:

   - All hardware components were automatically created during import in the "Unspecified DV location" folder in the "Locations" tab.
   - If a new bus line was created, the addresses of the new bus line may not match the COMOS data. In this case, a note is entered in the "Import/Export PCS 7" tab in the log area on the bottom.

3. Move the hardware components in the "Units" tab from the "Unspecified DV location" folder to the unit structure. The correct location is indicated by the PCS 7 structure.
4. Perform the "Assign fieldbus" engineering task.

   See section Editing hardware engineering using engineering tasks (Page 53).

3.13.6 Adapting the plant hierarchy

You can deviate from the default plant hierarchy. For more on this, see Assigning a plant hierarchy in COMOS (Page 36).

3.14 Reference

3.14.1 The "Automation Tree" tab

Objective of the "Automation Tree" tab

The "Automation Tree" tab is an intermediate step before data from PCS 7 are imported into the COMOS database or are exported from the COMOS database. Only objects that are relevant for the data transfer between COMOS and PCS 7 are displayed.

Important objects in the "Automation Tree" tab

- "PV Plant view"

  This folder is created by an import or the "Generate" step.
- "Comos_Lib Project library"
  - "Template-Container Templates"

    See Importing individual control unit types (Page 41).
  - "VAR Variants"

Use

The "Automation Tree" tab is a copy of the following structure: "Units" tab, "@AT Automation Tree" object.

The "Automation Tree" tab is read-only by default. To manage the Automation Tree, use the object "@AT Automation Tree" in the "Units" tab. The following attributes control the Automation Tree:

- "Read only" attribute

  Set or remove write protection for the "Automation Tree" tab.
- "Show the Automation Tree in a separate tree" attribute

  Displays or hides the "Automation Tree" tab.

3.14.2 Information in the "Data transfer - Generate/import" dialog

Objective

The "Data transfer - Generate/import" dialog is the user interface of the Automation Interface and can import the engineering data into the engineering project as well as the base project.

Log file

All events that occur during data transfer are logged in the status window. At the top edge of the status window, you can set which system messages are to be output:

- "Errors"

  Indicates an error. The import/export cannot take place if errors are detected.
- "Warnings"

  Shows information on errors that do not actually prevent an import/export but may lead to an unwanted result.
- "Messages"

  Logs the import/export process.

The entries in the status window are context-sensitive. The selection in the source text jumps to the respective place when you activate the entry by double-clicking on it in the status window.

Help on the "Data transfer - Generate/import" dialog

You can find additional information on the "Data transfer - Generate/import" dialog as follows:

1. Open the "Data transfer - Generate/import" dialog.
2. Click the "Help" button.

3.14.3 Navigator

"Units" tab

You configure the plant hierarchy (PH) in the "Units" tab and perform the software engineering.

Icon	Tab	Description
	"Units"	• Plant hierarchy The following nodes are also available here: • "@AT Automation Tree" Shows the interface objects of the "Automation Tree" tab. • "@Template Templates" Contains the imported control module library with the control module types from PCS 7. This folder is only created once the "Create templates" function has been executed.

The following objects can be found in this tab according to the example in this documentation:

Icon	Description
	Project
	Factory/building/production EN standard:
	Unit
	Part unit
	04 I&C (instrumentation & control)
	Position
	Control modules
	Parameter
	Signal
	Interconnection

"Locations" tab

The "Locations" tab is used to configure the objects for the hardware.

Icon	Tab	Description
	"Locations"	• Configured hardware

The following objects can be found in this tab according to the example in this documentation:

Icon	Description
	Project
	Factory/building/production EN standard:
	Building
	Floor
	Room
	Cabinet
	Station
	Rack
	Power supply
	Communication
	Central assembly

"Automation Tree" tab

The Automation Tree displays all objects relevant for the data transfer between PCS 7 and COMOS. This view is also used for monitoring data transfer between PCS 7 and COMOS.

Icon	Tab	Description
	"Automation Tree"	• PCS 7 project after import to COMOS • COMOS project before export to PCS 7

The following objects can be found in this tab according to the example in this documentation:

Icon	Description
	Project
	Master systems, e.g. Profibus, Ethernet
	Station, e.g. power supply, CPU
	Plant view
	Project library

3.14.4 The "Import/Export PCS 7" dialog

Objective

Transfer of engineering data between PCS 7 and COMOS.

Call

COMOS menu, "Plugins > Automation > PCS 7 interface > Import/Export PCS 7"

3.14.4.1 "Import" tab

Objective

After selection of a PCS7 project, you import the selected objects from PCS 7 to COMOS into the Automation Tree in this tab.

Reference

Control group	Control element	Description
"Source"	"PCS 7 project/library" field	Displays the PCS 7 project selected for import. You perform project selection in the Project selection dialog once you have clicked the "..." button.
"Source"	"XML file" field	Shows the XML file selected for import. You select an XML file in the File selection dialog once you have clicked the "..." button.

Control group	Control element	Description
"Filter"	"Plant hierarchy" option	Take the following objects into consideration in the selected source project: • All PCS 7 objects of the plant hierarchy. Following the import, the objects are created in the "Project > PV Plant View" node of the "Automation Tree" tab.
"Filter"	"Control modules" option	Take the following objects into consideration in the selected source project: • All PCS 7 control modules. Following the import, the objects are created in the "Project > PV Plant View" node of the "Automation Tree" tab.
"Filter"	"Control module types" option	Take the following objects into consideration in the selected source project: • All PCS 7 control module types. Following the import, the objects are created in the "Project > Project library > Templates" node of the "Automation Tree" tab.
"Filter"	"Hardware" option	Take the following objects into consideration in the selected source project: • All hardware objects from PCS 7. Following the import, the objects are created in the "Project" node of the "Automation Tree" tab.
"Target"	"Automation Tree project/library" list	In this list, you select the target project from the "Automation Tree" tab.
	"Import" button	Starts the import.

3.14.4.2 "Export" tab

After selection of a project in the Automation Tree, you export the selected objects from COMOS to PCS 7 in this tab.

Control group	Control element	Description
"Source"	"Automation Tree project/library" list	In this list, you select the source project of the "Automation Tree" tab to be exported.
"Filter"	"Plant hierarchy" option	Take the following objects into consideration in the selected source project: • All objects of the plant hierarchy in the "Project > PV Plant view" node.
"Filter"	"Control module" option	Take the following objects into consideration in the selected source project: • All individual control objects in the "Project > PV Plant view" node.

Control group	Control element	Description
"Filter"	"Hardware" option	Take the following objects into consideration in the selected source project: • All hardware objects in the "Project" node.
"Target"	"Automation Tree project/library" list	Select the PCS 7 target object from this list.
	"Import" button	Starts the import.

3.14.4.3 "Reengineering" tab

On this tab, reimported PCS 7 objects that have already been created in the COMOS project are matched and updated.

Control group	Control element	Description
"Source"	"Automation Tree project/library" list	In this list, you select the source project of the "Automation Tree" tab to be synchronized.
"Filter"	"Plant hierarchy" option	Take the following objects into consideration in the selected source project: • All objects of the plant hierarchy in the "Project > PV Plant view" node.
"Filter"	"Control module" option	Take the following objects into consideration in the selected source project: • All individual control objects in the "Project > PV Plant view" node.
"Filter"	"Hardware" option	Take the following objects into consideration in the selected source project: • All hardware objects in the "Project" node.
	"Match" button	Starts the matching process.

3.14.4.4 "Generate" tab

Objective

Use this tab to prepare the data of the COMOS project based on the rules of PCS 7 and buffer them in the Automation Tree. After generating the data, the COMOS data can be exported to PCS 7.

Reference

Control group	Control element	Description
	"Start object" field	Defines the hierarchy level as of which the stations are taken into consideration in the generation.
"Filter"	"Plant hierarchy" option	Takes into consideration the following objects as of the selected start object: • All objects of the plant hierarchy.
"Filter"	"Control modules" option	Takes into consideration the following objects as of the selected start object: • All control modules
"Filter"	"Hardware" option	Takes into consideration the following objects as of the selected start object: • All hardware objects
"Target"	"Station" column	Shows all stations below the selected start object.
"Target"	"PCS 7 project" column	Select an existing PCS 7 project as the target or create a new project here.
"Target"	"PCS 7 project path" column	Shows the project path to the selected PCS 7 project for each station.
	"Generate" button	Starts generation of the selected objects in the Automation Tree.

3.14.5 Reference to the mapping table

Objective

You can assign COMOS attributes to the copied and pasted PCS 7 control modules in the mapping table.

Call

Call of the mapping table:

- Properties of the control modules
- "Attributes > COMOS system" tab: "Open mapping table" button

Design

You can find additional information on this topic in the "COMOS Administration" manual, keyword "Mapping table".

Control element	Description
"Insert in template" icon	Saves the assignments in the template of the control module type used.
"Attribute" column	The attribute where the information is transferred.
"Description" column	Description of the set target attribute.
"Link type" column	To change the link type, double-click in this column and select another type in the list. • "Static" The linked value is not immediately accepted in the case of a static link. You initiate the use manually. A static link only serves as a base for comparison; the contents are not transferred. • "Dynamic" In the case of a dynamic link, the contents are always transferred from the source object to the attribute. • "Full dynamic" (recommended) All properties of the linked attributes are automatically transferred in both directions.
"Source object" column	The object from which the information is read. The object can be set as follows: • Via drag&drop from within the Navigator • In the navigation assistant See also section Reference to the navigation assistant (Page 101).
"Property" column	Contains the property being read out. If necessary, double-click in this column and select another property in the list.
"Value" column	Contains a preview of the information to be transferred.

3.14.6 Reference to the navigation assistant

The navigation assistant provides a graphical user interface for specifying the source object in the mapping table. It is used if the position of the source object in the unit tree is unknown.

You find additional information on this topic in the "Queries" manual, keyword "Calculation type: Navigation library expanded".

See also

Using the navigation assistant (Page 75)

Generic Excel import

4

4.1 Aim of the generic Excel import

The generic Excel import process involves the following functions:

- Importing hierarchy information and creating objects

  Available for system type: Base object (CDevice), document (Document)
- Importing or changing Values

  Available for system type: Attribute (Specification)
- Importing or changing standard properties

  Available for system type: Base object (CDevice)

4.2 Notes on structure of the Excel table

Structure of the Excel table

The Excel data must have the following structure:

- Header

  The Excel table must have a header.
- Data rows

  The values in all the cells of a column must have the same structure.

  Example: A cell contains the FullName of an attribute. In this case, all the values in the column must have the structure "<Tab>.<Attribute name>", for example

  (M001.T1, B01.T23, etc.).

Changes to the structure of the Excel table

The import rules for a "configuration" apply to a fixed structure in the Excel table. Performing the following actions will mean that the import rules are no longer appropriate for the structure of the Excel table:

- Creating or deleting columns
- Moving columns
- Renaming columns

4.3 Opening the "Generic Excel import" interface

To open the "Generic Excel import" plugin, follow these steps:

1. Open a project.
2. From the COMOS menu, select the "Plugins > Automation > Generic Excel import" command.

4.4 Select an Excel file

Procedure

1. Click the "Excel file" button and select a file.
2. In the "Excel worksheet" list, select an entry.
3. Enter a value in the "Excel start row" field.

   The "Excel start row" is the first row of the data area. All rows above are part of the header.
4. Click "Loading work sheet".

   The columns of the Excel file are displayed in the "Import data" area.
5. Select or clear the columns in the "Import data" area.

   Depending on what you select, the number of entries in the "Data assignment" working area will increase or decrease.

4.5 Defining import settings

Procedure

1. Optional: If a data assignment has already been saved as a configuration, click the "Configuration" button.

   See chapter Creating or editing the data assignment configuration (Page 114).
2. Use drag&drop to set an engineering object or the project node in the "Start object" field.
   - The objects are created or edited below the start object during the import process.
   - Based on the input in the "Start object" field, the "Base" field is also set. In this way, a root is also set for the Navigator in the "Data assignment" area.
   - If necessary, click on the "Remove pointer" button to the right of the "Base" field to remove this specification.
3. Use drag&drop to set a base object in the "Base" field.

   The base objects used in the "Data assignment" area are provided below this base object.

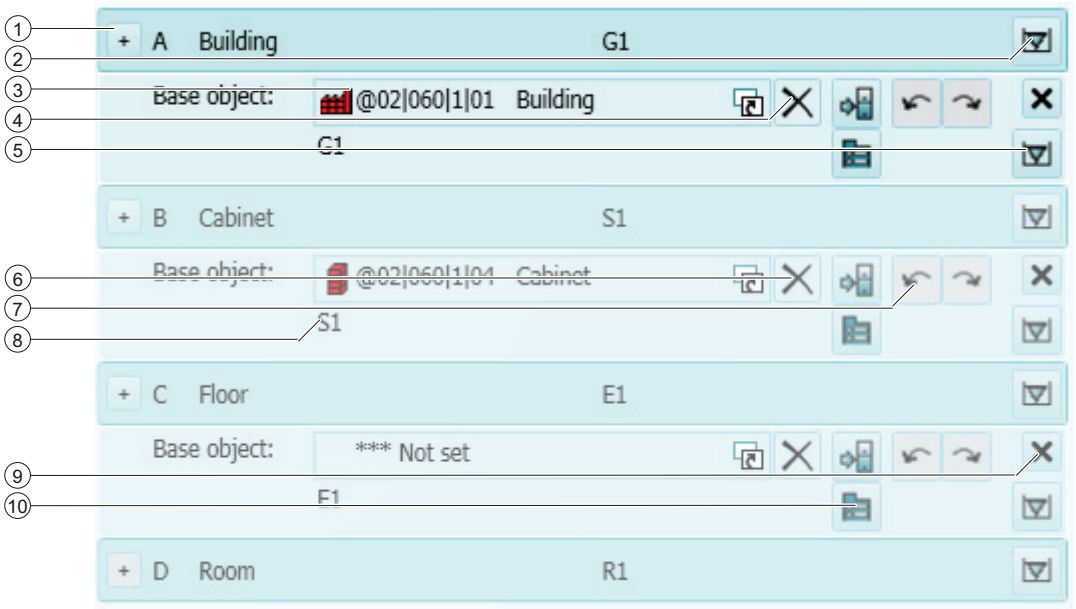
4.6 Interface reference for the "Data assignment" area

Requirement

Columns have been selected in the "Import data" area.

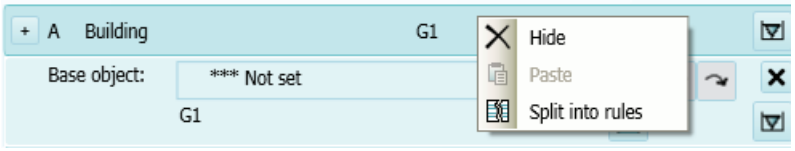
Control elements in the working area for rules

Once you have selected the "Loading work sheet" command, a header appears for each column in the "Data assignment" area. These entries have the following control elements:



1 Show rule/Hide rule
This row is referred to as "header of the rules".
2 Create new rule
3 Set the base object using drag&drop.
The base object must be below the object that is entered in the "Base" field.
4 Delete base object
5 Copy rule
A copy of the rule appears at the end of the rule list.
6 Edit rule
7 Calling the processing steps of the rule
8 Preview
9 Delete rule
10 Advanced rule options

Context menus



- "Hiding"

  Hides this entry.

  An entry can be shown again by selecting the check box in the "Import data" area.

- "Paste"

  To paste a processed rule, select the "Paste" command in the context menu of the header. Multiple selection is permitted.

- "Divide into rules"

  See chapter Divide into rules (Page 111).

4.7 Notes on importing hierarchies

Operation mode

The value in the Excel cell is read and split up, and a base object is assigned to each of the partial values. During the import process, an engineering object is created for each partial value. The result is that a hierarchy of engineering objects is determined and created on the basis of the Excel cell value.

Alternatively, an Excel cell receives exactly the information of a hierarchy level. In this case, the Excel cell value does not have to be split up. instead, there are various Excel columns which, collectively, contain the same hierarchy information.

Base objects for the administration of base data

COMOS recognizes base objects that are only used on the "Base data" tab. These base objects are used to administer and sort base data. These are base objects with the following properties, for example:

- "System" tab, "Block" placement option
- "System" tab, "Group" placement option
- "System" tab, "Structure" placement option

Base objects with these properties cannot be set in the "Data assignment" area.

Hierarchy of base data in COMOS

COMOS checks the hierarchy of the base objects below the object in the "Base" field during the import process. The following conditions apply:

- If the base objects have a hierarchy and if these base objects are used in the "Data assignment" area, the engineering objects are created hierarchically.
- If base objects are next to each other, the engineering objects will also be next to each other.
- If the base objects have a hierarchy that is only partially mapped in the "Data assignment" area, the gaps in the hierarchy are filled with dummy objects.
- It is permitted to use base objects in the hierarchy to administer the base data. These are ignored if the hierarchy is created on the engineering side.
- The sequence of columns in Excel does not have to match the sequence in the hierarchy. The columns are assigned in the "Data assignment" area of the hierarchy level.

Point to note when importing loops

If a new position diagram is created as part of an import operation, the components belonging to the loop are only visible when the "Place automatically" function is used.

The segments are visible from the first time you open a loop.

4.8 Data assignment for importing hierarchies

Requirement

Columns have been selected in the "Import data" area.

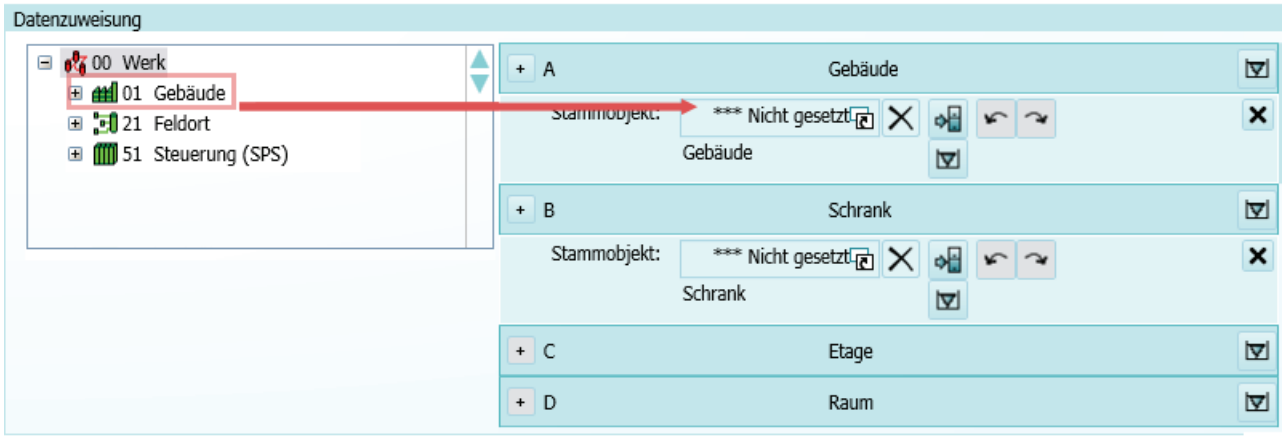
A base object was set in the "Base" field.

Procedure

To define a rule for reading a partial value, follow these steps:

1. In the "Data assignment" area, click "Create empty rule".
   - Optional: Click "Edit rule".
   - Select an entry from the "Select a function" list.
   - Define the rule for the selected entry.

     See also chapter Edit rule/Splitting (Page 111).
   - Confirm with "OK".
2. From the "Data assignment" area in the Navigator, move a base object to the "Base object" field using drag&drop.



Note: the "Find base object based on name" option changes how the "Base object" field is evaluated. Refer to section Extended rule options (Page 113).

Permitted entries in the "Base object" field for creating new objects are:

- Base objects (CDevice)
- Documents

3. To create an additional rule for the same Excel column, click "Create empty rule" again.
4. Edit the next rows in the "Data assignment" area using the same procedure.

Result

The "Data assignment" area shows an example of what effect the rule has on the first cell in the column. During the import process, the rule is applied to all the cells in the column. Engineering objects are created or found on the basis of the base object that is set. The engineering objects that are created are given a name in accordance with the name displayed at the base object.

4.9 Data assignment for importing a value

Requirement

Columns have been selected in the "Import data" area.

A base object was set in the "Base" field.

Procedure

To define a rule for importing a "Value" attribute property, follow these steps:

1. In the "Data assignment" area, click "Create empty rule".
   - Optional: Click "Edit rule".
   - Select an entry from the "Select a function" list.
   - Define the rule for the selected entry.

     See also chapter Edit rule/Splitting (Page 111).
   - Confirm with "OK".
2. From the "Data assignment" area in the Navigator, move a base object of an attribute to the "Base object" field using drag&drop.

   The checks in the "Extended rule options" dialog are deactivated because only the default properties are permitted.
3. To create an additional rule for the same row, click "Create empty rule" in the "Data assignment" area again.
4. Edit the next rows in the "Data assignment" area using the same procedure.

Result

A search for the CDevice owner is performed for the base object of the attribute. A search is performed for the engineering object associated with this owner, and the attribute is updated here.

4.10 Data assignment for importing changed properties

Requirement

Columns have been selected in the "Import data" area.

A base object was set in the "Base" field.

Operation mode

To change a property, each object must be listed twice in the Excel table:

- Entry with the previous name

  Below, this Excel column will be referred to as: Name column.

  The object is identified with this entry.

- Entry with the property to be changed

  Below, this Excel column will be referred to as: Property column.

  Properties that can be changed are:

  – Name

  – Description

  – Label

Procedure

To define a rule for importing changed properties, follow these steps:

1. Editing the "data assignment" of the name column
   – In the "Data assignment" area, select the name column entry. Click "Create empty rule".
   – Optional: Click "Edit rule" and edit the dialog. See also chapter Edit rule/Splitting (Page 111).
   – From the "Data assignment" area in the Navigator, move a base object (CDevice) to the "Base object" field using drag&drop.

     Do not change the default settings in the "Extended rule options" dialog.
2. Editing the "data assignment" of the property column
   – In the "Data assignment" area, select the property column entry.
   – Optional: Click "Edit rule" and edit the dialog. See also chapter Edit rule/Splitting (Page 111).
   – From the "Data assignment" area in the Navigator, move the same base object (CDevice) as in the previous step to the "Base object" field using drag&drop.
   – Open the "Extended rule options" dialog. See also chapter Extended rule options (Page 113).
   – Select an entry in the "Target property" list
3. Edit the next rows in the "Data assignment" area using the same procedure.

Result

An engineering object is searched for in accordance with the value in the name column and the set base object. The property is updated here in accordance with the value in the property column.

4.11 Divide into rules

Procedure

1. Select one or more rules.
2. Select the context command "Divide into rules" in the header.
3. Enter a separator.
4. Confirm it.

Result

The value of the Excel cell is automatically divided into several partial values. The value of the Excel cell is searched for the separator for this purpose. A rule is created for each hit.

4.12 Edit rule/Splitting

Opening the "Splitting" dialog

The "Splitting" dialog is opened in the "Data assignment" area using the "Edit rule" button:

Purpose

The "Splitting" dialog is used to read out only a part of the contents of an Excel cell.

If each Excel cell contains exactly the text required for that cell during the import, the "Splitting" dialog is not needed.

"Select a function" list

- "Regular expression"

  Separates the data using a regular expression. Regular expressions as per .NET are permitted. An Internet search using the keyword ".NET Framework Regular Expressions" will help you find more information about this.

  – "Regular expressions (examples)" field:

    The "." character acts as a wildcard for any one character.

    The "*" character acts as a wildcard for any number of characters.

    The "\d" characters stand for any one digit.

  – "Predefined expressions" field

    Includes application cases often needed for regular expressions.

- "Split character"

  You may only complete either the "From front" field or the "From rear" field.

  Separates data at the point where the specified character is found.

- "Substring action"

  You may only complete either the "From front" field or the "From rear" field.

  Separates the data after the specified number of characters.

"Predefined expressions" list

This list includes examples for regular expressions. The examples can be applied with "Paste".

4.13 Extended rule options

Extended rule options

Available for: Base object (CDevice), document (Document). The entries in the dialog cannot be changed if an attribute is set.

- "Find base object based on name"

  Purpose: Name-based assignment of a base object to a rule.

  Starting at the specified node, the Excel content determined by the rule is interpreted as the name of a base object that is located below the node.

  Example: different base objects exist for signals. Due to the high overheads involved, it is not practical to set up a separate rule for each signal. In this case, the following procedure is possible:

  – At the rule, you set the owner base object that has the different base objects of the signals as subobjects.
  – You enable the extended option "Find base object based on name"

- "Target property"

  The selection in the "Target property" list can be evaluated both for newly created objects and when existing objects are changed.

  – " Standard"

    For base object and document: The imported or calculated string is interpreted as a name. If an object with this name is found, it is processed. If no object with this name is found, a new object is created.

    For attribute: The imported or calculated string is interpreted as a value. Note: not all attributes use the value as a standard property. The "Edit field" attribute can be used for testing.

  – "Name"

    Changes or sets the name of the object.

  – " Description"

    Changes or sets the description of the object.

  – "Label"

    Changes or sets the label of the object.

4.14 Information on editing rules

The following information will make it easier for you to work with rules:

- The history of editing steps for a rule remains available for as long as the "Generic Excel import" is open. You can call up the editing steps using the "Forward" and "Back" arrow buttons.
- Provided the "Generic Excel import" tab is open, you can hide a column in the "Import data" area without losing the editing steps history for the associated rules. If you show the column again, the rules will appear and the editing steps can be called up.

4.15 Creating or editing the data assignment configuration

Creating a new configuration

1. Click "Configuration...".
2. Click "New".

   A new entry appears in the list on the left.

   Configurations are stored in the database as follows:

   "@System System settings> @Profiles > <User> > GenericExcelImport User settings"
3. Click the button containing the cross to close the window.

Using a configuration

1. Click "Configuration...".
2. Select a configuration from the list on the left.
3. Click "Load".

Editing a configuration

1. Change the rules in the "Data assignment" area.
2. Click "Configuration...".
3. Select a configuration from the list on the left.
4. Click "Save".

Deleting a configuration

1. Click "Configuration...".
2. Select a configuration from the list on the left.
3. Click "Delete".
4. Click the button containing the cross to close the window.

Renaming a configuration

1. Click "Configuration...".
2. Select a configuration from the list on the left.
3. Click "Rename".
4. Click the configuration in the list on the left again.
5. Overwrite the name.
6. Click "Save".

Export configuration

1. Click "Configuration...".
2. Select a configuration from the list on the left.
3. Click "Export".
4. Enter a name.

   The configuration is saved as an xlm file.
5. Click "Save".

Import configuration

1. Click "Configuration...".
2. Click "Import".
3. Select a file.

   The selected configuration then appears on the left in the list.
4. Select the imported configuration from the list on the left.
5. Click "Load".

4.16 Conducting the import

Checking the import preview

1. Complete the following field:

   "Import settings" group, "Preview line" field
2. Click "Show preview".

   The "Preview window" dialog, in which an import is simulated starting from the object in the "Start object" field, opens.

Initiating import

To import the data into the open working layer, follow these steps:

- Click "Import".

To import the data into a new working layer, follow these steps:

- Click "Import into working layer".

  The new working layer opens automatically.

Effect on inventory data

If the import takes place several times or if an imported takes place on top of existing data, the data from the Excel table are always added to the COMOS DB or the existing data in the COMOS DB is overwritten.

If an object already exists, it is overwritten.

There is no synchronization: Data in the COMOS DB are not deleted if they are missing in the Excel table.

Creating structure objects

Objects with the "Structure" creation option are taken into account.

If a base object is prepared as element with the "Structure" creation option, the following applies: If the element was created with the property "Virtual: Off", it is automatically created when the parent object is imported.

This way a COMOS-specific object structure can be automatically created during the import.

Process visualization via OPC client 5

5.1 Objective

OPC = Open Process Control

An OPC connection is used for transferring and displaying values (read mode). This means that it is a unidirectional connection. It does not involve an import in the strict sense of the word, because the displayed values are not saved initially.

OPC servers are primarily used for process visualization purposes.

5.2 Basic principle of visualization

The client queries the current values of signals from the server; these are then saved on the "OPC" tab of the signal.

The OPC connection is implemented in COMOS as follows:

- An OPC server must be available; this is not included in the COMOS scope of delivery.
- COMOS has its own OPC client.
- A script is entered at an attribute, which automatically starts the COMOS OPC client as soon as the displayed value (DisplayValue) of the attribute is accessed.

5.3 Examples of supported OPC servers

COMOS has an OPC client that can address any OPC server.

Examples of software with the OPC Server:

- Siemens: WinCC
- ETM: PVSS II

5.4 Project attributes: Specifying the OPC Server

Open: Properties of the base project, "OPC" tab

Control element	Description
"Server name" field	Is only evaluated when AutoStart is turned off. If the OPC server and COMOS are running on the same computer, you do not need to specify a server name. Otherwise, prefix the computer name with "\\" (e.g. \\Comp007).
"Computer name" field	Is only evaluated when AutoStart is turned off.
"Autostart" option	Deactivated: a registration form is displayed on starting the OPC client. Enter the server name and computer name here. Activated: the form does not appear and the necessary information is taken from the "OPC" tab in the project.

5.5 Attributes at the engineering object: Calling the OPC Client

Attributes at the engineering object: Calling the OPC Client

Example:

Open: "S01 > @02 > 050 Signals > 00 Signal, general > IEC 1175", "OPC" tab

Control element	Description
"OPC variable name"	Full name of the signal
"Online value"	• Attribute properties, "Edit mode" attribute is "Graphical user interface with script" • Displays the current value of the signal transmitted by the OPC Server • Is linked via GetDisplayValue

Function GetDisplayValue()

Script function at the "Online value" attribute for determining the output value of the attribute.

Return value: String

Open function: "S01 > @10 > BAS > 0 > SIG > OPC > VarValue", "Script" tab

5.6 Technical implementation / Application

Technical implementation / Application

Access to the OPC Client is conducted automatically as soon as the VarValue attribute is accessed. The mechanism only works in the engineering view.

1. Open the engineering object properties and display the tab where the attribute is located.
2. Unfold the Navigator until the attribute is visible in the engineering view
3. Open the report on which the attribute is output

On the first access, the GetDisplayValue Script block starts.

An OPCItem object is created and the name of the to be linked OPC variable is transmitted. The COMOS OPC client starts.

It is analyzed whether an item with the entered name already exists; if it does not, it is created. The item is added to the group of the active variable. The current value is returned and displayed at the attribute.

The server now monitors the variable group generated by COMOS. The COMOS OPC client is informed when a value from this group changes. Thereupon, the OPC Client induces the update of the attribute.

EPLAN 5.x (EXF import/export) 6

6.1 Background information

6.1.1 EPLAN structures compared to COMOS structures

Overview

EPLAN organizes its data as follows:

"Normal" base data and article base data

This splitting roughly corresponds to the COMOS scheme. COMOS also contains base data in the form of templates for engineering data (EPLAN: "project data"). The article base data roughly corresponds to the manufacturer catalog in COMOS.

In EPLAN there is a strict separation of "normal" base data and article base data.

In COMOS, the manufacturer catalog is managed within the normal base data. For that reason, "normal" base data from EPLAN is not imported into COMOS. You only import article base data into COMOS.

There is only one form of import for article base data. There is no interface for exporting article base data from COMOS to EPLAN.

Symbol files

A symbol file contains all symbols from one of the relevant seven symbol types within EPLAN. Therefore, a component can appear in a maximum of seven different display types.

- Variants in EPLAN symbols

  Position changes (rotations) are controlled by variants. The rotated symbol is not calculated. You can create a separate symbol for up to four different rotation angles.

  You can use and process every symbol variant during the import.

  Rotated symbols cannot be processed during the export, since there is no corresponding form for rotated symbols in EPLAN. If you export a rotated variant to EPLAN, the rotation will be ignored and the basic symbol exported instead.

  Recommendation: If you intend to export to EPLAN, only work with non-rotated symbol variants.

- Separate management of symbols in EPLAN

  In EPLAN the symbol files are kept strictly separate from the base data and the project data. Within the project data there is only a text reference to a symbol, which is then loaded during runtime.

  In COMOS, the symbols are part of the base objects and thus also part of the engineering objects as a result of inheritance.

Plot frames

A plot frame is used to print ("plot") information. A plot frame corresponds to a master report in COMOS.

Document management is relatively limited in EPLAN; there is only one plot frame or there are just a few plot frames.

You can use as many reports as you want in COMOS; you do not necessarily have to work with master reports. You have the option to directly create a report, which you can then print out.

Forms

Forms take in information and display it. They are used within a plot frame and are then called plot forms. Forms are not simply evaluation reports from a COMOS point of view.

The EPLAN default forms are:

- Terminal diagram
- Cable plan
- Terminal strips overview
- Cable overview
- Terminal connection diagram
- Parts list
- Device list

- Terminal parts list
- Control cabinet design
- Table of Contents
- Title page/cover page
- PLC overview

Compared to EPLAN, COMOS has more options for handling information. For example, you can collect data in queries that is completely separate from any type of documentary display. COMOS provides a graphic representation of the data in the reports.

Project data

EPLAN projects, i.e. engineering data, only exist in the form of documents ("pages"). These pages contain all engineering details and also anything from one to seven references (links) to symbol files. There is no separate type of data management such as there is in COMOS in the form of objects.

COMOS also features a document-oriented view of the engineering data in the form of reports, but COMOS reports and EPLAN pages do not match up in terms of their functions. See also the section titled "EPLAN pages compared to COMOS reports (Page 124)".

Binary files

Binary files can be managed within EPLAN. COMOS can manage binary files too, but it cannot import them via the EPLAN interface. You must import the files into COMOS manually.

File endings

- *.exf

  EPLAN eXchange format. An EPLAN project is saved and managed in a database. This database consists of multiple single files. The EXF format provides a complete EPLAN project as a single file.

  Operation in EPLAN: *drawings "Data exchange" > EPLAN -> EXF export > Circuit diagram: creates an EXF file.

- *.sym

  Symbol files; the format corresponds to the symbols of the base data in COMOS.

  Operation in EPLAN: *symbols (.SYM) "Data exchange" > EPLAN -> EXF export > Plot frame: creates an ASC file for each symbol database.

- *.asc

  Article base data, is created from the converted SYM files.

- *.skg

  Eplan plot frame, this is approximately like a report template.

  Operation in EPLAN: *plot frame (.SKG) "Data exchange" > EPLAN -> EXF export > Plot frame: creates an EXP file for each plot frame.

- *.exp

  Eplan plot frame, is created from a converted SKG file.

- *.sk* (except *.skg)

  Forms (.SK*) depends on the form type. Operation in EPLAN: "Data exchange" > EPLAN -> EXF export > Forms: creates an NTO file for each form.

6.1.2 EPLAN pages compared to COMOS reports

Page numbering

The engineering data for EPLAN projects consists of documents or pages. The engineering information is primarily identified via the designation of documents, which is why this property is much more significant in EPLAN than it is in COMOS. See also the section titled "Preparing a COMOS engineering project (Page 127)".

Page type

EPLAN pages always have an entry in the "Type" field, which refers to a page type.

ID code	Page type	Description	Counterpart in COMOS
A	Circuit diagram	Logical, Interactive	Functional plan
B	Free graphic	Graphic, Interactive	Interactive report
C	Control cabinet design	Graphic, Interactive	Interactive report
D	Plot frame generation	Graphic, Interactive	Interactive report
E	Title page/cover page	Graphic, Interactive	Interactive report
J	Table of contents	Graphic, Automatic	Evaluation report
K	Terminal diagram	Graphic, Automatic	Evaluation report
L	Terminal parts list	Graphic, Automatic	Evaluation report
M	Terminal connection diagram	Graphic, Automatic	Evaluation report
N	Cable plan	Graphic, Automatic	Evaluation report
O	Parts list	Graphic, Automatic	Evaluation report
U	Order list	Graphic, Automatic	Evaluation report
P	Device list	Graphic, Automatic	Evaluation report
Q	PLC page	Logical, Automatic	Functional plan
R	Terminal strip overview	Graphic, Automatic	Evaluation report
S	Cable overview	Graphic, Automatic	Evaluation report
T	PLC overview	Graphic, Interactive	Interactive report

EPLAN "Type" and EPLAN "Form"

The EPLAN "Type" often corresponds to the EPLAN "Form".

"Type" and "Form" do not have a general counterpart in COMOS, since COMOS requires you to make clearer distinctions. In COMOS, a combination of three different factors equates to the EPLAN "Type" and "Form":

- "Drawing type"; "Symbol Type" in the report template

  EPLAN pages are converted into one of the DETAIL drawing types. These drawing types are created especially for EE. Evaluation reports do not have any drawing type.

- "Report template"

  The report templates are made up of a master report and a sub report. The master report is generated from a plot frame and the sub report from a form.

- Document base object

  Document base objects are created in "@O Documents > EXF EXF import". Document base objects for EXF have the "EXF settings" tab and the following dialog fields there:

"Page type" (COMOS field):

- EPLAN page
- Free graphic
- FG Control cabinet design
- FG table of contents
- Plot frame generation
- RS scheme
- Terminal module holder overview
- SAPIM
- NCRS
- FG terminals plan
- FG terminals parts list
- FG terminals connection plan
- FG cable plan
- FG parts list
- FG device list
- EPLAN page for PLC
- Terminal strip overview
- Cable overview
- Title page/cover page
- PLC overview

"Form" field:

The form being used is automatically entered into this field when the EPLAN project data is imported. The text is transferred back to EPLAN when exporting and EPLAN assigns a form on the basis of the text.

Many pages in EPLAN have a form; whether or not an EPLAN page possesses a form does not depend on whether this page is interactive.

Page size

EPLAN pages do not use DIN sizes. The page sizes must be transferred as well when importing EPLAN pages.

In the COMOSDB, there are a number of report templates that have already been prepared with valid EPLAN page sizes. These report templates are labeled with the supplement "EPlangr".

Multi-page documents

In EPLAN there are no multi-page documents. Each document thus comprises exactly one page. The affiliation of the pages can be determined on the basis of the numbering. If pages are generated automatically via a form, multiple separate pages are generated as required.

In COMOS, there are multi-page documents.

Report templates for EXF: `EXFConformable`

The script option ExfConformable can be set in the options script of a report template.

Effect:

- The texts of the labelling segments can no longer be moved but instead appear at specific positions. This position of the texts is controlled via: marking a segment, right-click, context menu "Properties" > options group "Label position".
- Texts of components/devices, etc., can no longer be moved but appear at specific positions.
- The grab points are deactivated.

6.1.3 Overview of exporting COMOS data to EPLAN

"Normal" COMOS data can also be exported to EPLAN in principle, but this data is then placed as images or free graphics. Thus this data can no longer be changed or edited within EPLAN. In practice, it is virtually impossible to organize the original COMOS data in such a way that a functional and consistent EPLAN project can be generated using it.

If you nonetheless wish to work further with EPLAN, then proceed as follows:

- Create a special EXF project in COMOS.

  See also the section titled "Preparing a COMOS engineering project (Page 127)".

- Export all required base data in EPLAN:

  Article base data; Symbols; Plot frame; Forms

  See the EPLAN documentation, you can find a schematic overview of these procedures in exporting article base data from EPLAN, p. 13-181, Exporting symbols from EPLAN, p. 13-183, Export plot frames from EPLAN, p. 13-187, Exporting forms from EPLAN, p. 13-189.

- Import the EPLAN base data to COMOS.

  See also the section titled "Importing EPLAN data (EPLAN -> COMOS) (Page 131)".

- Export an engineering project as EXF in EPLAN.

  See the EPLAN documentation, there is a schematic overview of this procedure in the section titled "Exporting projects from EPLAN (Page 138)" too.

- Import the EXF data to COMOS.

  See also the section titled "Importing EPLAN data (EPLAN -> COMOS) (Page 131)".

- Continue processing the data in COMOS.

  See also the section titled "Working in COMOS with (imported) EXF data (Page 143)".

- Export the modified engineering data as EXF data in COMOS.

  See also the section titled "Exporting from COMOS (COMOS -> ExF) (Page 146)".

- Import the EXF data to EPLAN:

  See the EPLAN documentation.

6.2 Preparing a COMOS engineering project

The COMOS project must be linked with the following template: "@J Project > @X Project settings EXF".

This project template has the following specialties:

- An "EXF" attribute tab is created in the project. This "EXF" tab is created in the base data in "@Y Catalog attributes > EXF EXF import > 04 Project settings EXF".

  Purpose: EPLAN projects have project information. When importing to COMOS, this project information is read from the header of the exf file and written to the "EXF" tab. This data is evaluated again during an export operation and is also written into the header of the exported EXF file. See further below for information on the "EXF" tab.

- A unit system is prepared in the context menu.
- The default document folder is prepared in the context menu.
- A location system is prepared in the context menu.

- The unit system, the default document folder, and the location system originate from "@J Project > @X Project settings EXF.
- For information on the prepared context menu, see the section titled "Working in COMOS with (imported) EXF data (Page 143)".

The EXF tab

- Page numbering

  EPLAN projects (i.e. engineering data) fundamentally only exist in the form of documents ("pages"). The engineering information is primarily identified via the designation of documents, which is why this property is much more significant in EPLAN than it is in COMOS.

  COMOS documents also have a "page", but this property actually contains only a page number. The complex page numbering used in EPLAN is expressed in COMOS through the hierarchical structure of the engineering data.

  Depending on the EPLAN page numbering, the engineering data created within COMOS is set up differently in an import operation:

  – Overall numbering

    The documents have a sequential number. This corresponds to the COMOS procedure of numbering documents sequentially within a document group. Within EPLAN, "subpages" (2a, 3B, 3.1, 3.8, etc.) are also permitted in such a case. Overall numbering within EPLAN extends from 1 to 99999. Within a unit, counts depend on the overall numbering criteria.

    Import: The newly generated reports are only created on the "Documents" tab.

  – DIN: Unit (descr. location)

    Apart from the page number, the unit is also used for identification. The location is added as an additional description but does not play any role at all for identification. In practical terms, therefore, the option is identical to "DIN: unit only".

    Import: The newly generated reports are only created on the "Units" tab; the unit structure is generated accordingly.

  – DIN: Unit and location

    Apart from the page number, the unit and location are also used for identification. EPLAN: Locations of the same name can also occur with different units; in such cases it does not involve the same locations. The numbering always starts at 1 within the locations.

    Import: The newly generated reports are only created on the Units tab, but locations are also created underneath the units. This involves what is known as a "mixed structure" from the COMOS point of view.

  – DIN: Only location

    Apart from the page number, the location is also used for identification.

    Import: The newly generated reports are only created on the Locations tab; the location structure is generated accordingly.

  – DIN: Only unit

    Apart from the page number, the unit is also used for identification. In practical terms, therefore, the option is identical to "DIN: unit only (restr. location)".

    Import: The newly generated reports are only created on the "Units" tab; the unit structure is generated accordingly.

  – KKS

EPLAN: The KKS label (German power station labelling system) made up of two characters is applied as the superordinate sort criteria. The unit comes after that.

Import: The newly generated reports are only created on the "Units" tab; the unit structure is generated accordingly.

- "Type of labelling"

  This determines how the devices are labelled on the documents (or reports, as applicable). EPLAN and COMOS work in a sufficiently similar way to each other in this respect. Example: If label type "DIN: unit and location" is selected for the device, then the following text is output on the report at the device: "=Anl1+Ort1|Kennzeichen". In the case of "Anl1", etc., this of course involves a placeholder that is filled by the corresponding text in the report. There are the following labelling types:

  – "Device"

    Import: The device label is created accordingly during the import operation. Any subsequent changes to this entry no longer have any effect on existing COMOS data.

    Export: Is evaluated correspondingly in EPLAN. Please note: This setting must be reconciled with "Page numbering" in accordance with the rules in EPLAN! The rules currently applicable can be found in the EPLAN manual.

    Example (valid as of the date of creating this manual):

    "Page numbering" "1 overall numbering" allowed:

    "1 Overall: Without page prefixes" / "2 Overall: With page prefixes"

  – "Terminals"

    Import: The EPLAN setting is used on the "EXF" tab, but is not evaluated in COMOS.

    Export: Is evaluated correspondingly in EPLAN.

  – "Potentials"

    Import: The EPLAN setting is used on the "EXF" tab, but is not evaluated in COMOS.

    Export: Is evaluated correspondingly in EPLAN.

  – "Project data"

    General details.

    Import: The EPLAN setting is used on the "EXF" tab, but is not evaluated in COMOS.

    Export: Effect in EPLAN unknown.

Other specailties at the project level

COMOS project options, "Options" tab:

Do not use "blanks" for units of the same type. This setting cannot be read within EPLAN.

6.3 Importing EPLAN data (EPLAN -> COMOS)

6.3.1 Importing EPLAN article base data to COMOS

Exporting article base data from EPLAN

You can find details of the applicable procedure in the EPLAN documentation.

The basic procedure is as follows:

1. Click the "Article > Management" command in the EPLAN main menu.
2. Click "Data exchange > Export" in the next window.
3. Mark the "Individual part" option on the "EPLAN item management - Export" tab. All other options can be left at the defaults. The article base data is exported as an ASC file.

Importing article base data to COMOS

 Select the "Administrator > Base data > Import ECAD Components" menu command.

The article base data must be imported before the engineering data in all cases, since otherwise the article base data cannot be allocated to these objects when importing the engineering data.

Often the article base data is only imported once, since as a rule the base data remains constant for a certain period of time.

- "File"

  Here the file type must be changed to ASC. After that the EPLAN file with the article base data is selected.

- "Target"

  In the COMOSDB, the following object must be set here using drag&drop: "Import > EXF EPlan > SYSTEM Special symbols > @A Article/Manufacturer catalog"

- "Read accessories"

  This option is not evaluated in the case of ASC files.

  "ASCII-ANSI character set conversion"

  This option is not evaluated in the case of ASC files.

- "Log all"

  If this option is activated, then all steps of the import operation are logged, otherwise only any errors that occur.

COMOS structure of the imported data

The main level of the base data already exists under the "@A Article/Manufacturer catalog" node. Base objects are created at this first level and are assigned attribute tabs on the basis of the imported data.

First of all, a level is created for the manufacturer and the articles follow below it. Currently, the structure taken from EPLAN is then as follows:

```
Import
|- EXF EPlan Import/Export
   |- System special symbols
      |- @A Article/manufacturer catalog
         | 0 cannot be identified
            |-...
         |- A Assemblies
            |- Bosch
               |- 123abc article1
               |- ...
            |- Siemens
               |- 456abc article1
               |- ...
         |- and so on
```

6.3.2 Importing EPLAN symbols to COMOS

Exporting symbols from EPLAN

 You can find details of the applicable procedure in the EPLAN documentation. The basic procedure is as follows:

1. In the EPLAN main menu: "Data exchange > EPLAN > EXF export".
2. In the following window, click on "Export > Symbol files".

   Symbol files have the extension "SYM" and can be written as text files with the extension "ASC".

Importing symbols into COMOS

The Import/Export dialog is available in the "PlugIns" iconbar.

The symbol files must be imported before the engineering data in all cases, otherwise the symbols cannot be allocated to these objects when importing the engineering data. Please note: It may be necessary to call this functionality several times if the desired EPLAN project addresses several symbol files.

Often the symbols are only imported once, since as a rule the symbol libraries remain constant for a certain time period.

If the base data structure of a symbol file is to be updated, then you must delete the corresponding base data branch in COMOS and reimport the symbol file.

The "Symbol import" tab

- "File"

  Here you can select the ASC file with the symbols.

The EXF tab in the case of symbol objects

- "Component type"

  This field offers a standard table of all possible EPLAN components. This list is also gets the cross-reference of the display. The component type also determines the allocation of the class, subclass and "Special symbol". See below for more information.

- "Caption text"

  In EPLAN texts that originate from the "Electrical value" and "Additional text" attributes are output underneath a device.

  The attributes "M11 Electrical indicator" and "M12 Additional field" are, therefore, created for it on the COMOS "EXF" tab.

  These attributes can be joined individually with other attributes by means of links.

  Examples:

  – In the case of a motor, with the capacity
  – In the case of a fuse, with the turn-off capacity.
  – In the case of other objects, with the description of the device (OwnDescription)

- "Article data"

  In EPLAN up to 10 article numbers can be allocated to the components in the project data.

  Currently, the first article number is read in COMOS when importing the project data and written in the device of the assigned symbol object in the first field of the "Article data". See also the section titled "Importing project data (Page 138)".

COMOS structure of the imported data

A symbol file is transferred to a COMOS structure during the conversion process:

1. Central import node "Import > EXF EPlan"

   Please note: The import to the base project is made in all cases, even when an engineering project has been opened! You must therefore have the corresponding function rights.

2. Importing symbol files

   One import per symbol file. The symbol files are created as base data; thus a separate branch is created for each symbol file. If all seven formats in which the symbol can be displayed are used within EPLAN, then seven corresponding symbol files must be available and you must perform the import to COMOS seven times.

   However, in COMOS you are not restricted to seven symbol files. Underneath "Import > EXF EPlan" you can create not just seven branches (= symbol files), but as many as you wish.

3. One base object per symbol

   One base object is created for each symbol. A symbol holds up to 512 symbols. As a rule, placeholder objects are created for the symbols that have not been used yet. The "Name" of the symbol base objects runs from 0 to 511.

   The following steps are then carried out in the base objects of the symbols:

- Determine class / subclass

  The Class / Subclass is set accordingly:

  – Contacts (normally open, normally closed, etc.), component type from [0 - 49]: Class "Element", subclass "None"

  – Coils, component type from [50 - 99]: Class "Device", subclass "Contactors/relays"

  – Terminals, component type from [100 - 149]: Class "Element", subclass "Terminal"

  – All other objects: Class "Device"

- Transferring the tabs:

  – "EBD" Mounting data

  – "GD" Geometry data

  – "HSD" Manufacturer data

  – "LFD" Supplier data

  – EXF

  – SYS system

Note

Nothing is created on the "Symbols" tab at this level of the base data. The symbol display is only done at the level of "variants".

1. One element object per "variant"

   Up to four elements are created underneath the base object. Each of the four elements thus displays one of the variants, with which symbols can be displayed in rotated form within EPLAN.

   A base data level is created for the variants in all cases, even if there is only one variant.

   The base data structure looks like the following if configured correctly (example):

```
Import
|- EXF EPlan import/export
 |- DIC_WUPD symbol file
 |- ...
 |- 259 MMT motor with thermistor
 |- 0 MMT motor with thermistor
 |- 1 MMT motor with thermistor
```

   and so on.

   The following steps are then carried out in the base objects of the variants:

   – Allocation "Special symbols"

     Contacts and terminals are identified on the basis of the label abbreviation (S, SV, etc.). A base object with a matching name is searched for in the "Import > EXF Eplan import/export > SYSTEM Special symbols" base data branch. If a suitable base object is found, a base object link is set on the "System" tab of the variant object.

     This allocation to the objects in SYSTEM special symbols is also needed in order to set up a contact mirror.

     The base object link interrupts the hierarchical inheritance. The element object now inherits from the base object to which the reference (link) points. Exception: Symbol script, see the following.

     If no base object link is set, then the variants inherit the attribute tabs transferred from @A Article and the class/subclass from the symbol base object specified in 3.

   – Transferring the EPLAN symbol to the "Symbols" tab

     A symbol script is generated from the ExF symbol file data and stored in COMOS at the base object of the relevant element on the "Symbols" tab.

     This is also done if a base object reference had been set!

   – Setting up a naming convention

     If there is corresponding data in the symbol file, a corresponding text mask is generated on the "System" tab.

     This does not occur in the case of a base object link.

   – Creating connectors

     In COMOS, symbols must have connectors so that they can be connected. This is done by generating the corresponding connectors on the "Connectors" tab.

     This does not occur in the case of a base object link.

6.3.3 Importing plot frames

Export plot frames from EPLAN

You can find details of the applicable procedure in the EPLAN documentation. The basic procedure is as follows:

- In the EPLAN main menu: "Data exchange > EPLAN > EXF export".
- In the following window, click on "Export > Plot frame".
- The plot frame is written to an EXP file when it is exported.

Importing plot frames to COMOS

 The Import/Export dialog is located in the "PlugIns" iconbar. Plot frames are created within COMOS in the form of master reports.

The plot frames must necessarily be imported before the forms, since otherwise it will not be possible for the plot frames (or in COMOS, the master report) to be allocated to these objects when the forms are imported.

Plot frames are often only imported once, since as a rule the templates remain constant for a certain time period.

On importing a plot frame, the date and user are taken over from EPLAN. This means that, instead of issuing a new timestamp for the master report, the master report takes over the data of the plot frame.

The Plot frame tab

- "File"

  Here you select an EXP file that contains the plot frame.

- "Forms to"

  Plot frames are converted into report templates within COMOS. Here you stipulate where the corresponding report templates are to be stored. As is usual within COMOS, the report templates can only be created within the base project.

  A document group has already been prepared within the COMOSDB. When the "EPLAN data exchange" tab is opened, this document group is entered as the default. However, another document group can be set instead by using drag&drop.

- "PLC overview"

  No special plot frames are used within EPLAN for page type "PLC overview". In COMOS, special reports are used for PLCs.

  The simplest way to synchronize EPLAN and COMOS here is to import the same EPLAN plot frame twice and to import the plot frame with the "PLC overview" option once. Effect:

  – PLC overview deactivated: The Master Report that was created is given the "Detail" drawing type.

  – PLC overview activated: The created Master Report is given in the "Detail_2" drawing type. In addition, the document objects and the CRP files receive the suffix "PLC" at the end of the name.

6.3.4 Importing forms

Exporting forms from EPLAN

You can find details of the applicable procedure in the EPLAN documentation. The basic procedure is as follows:

In the EPLAN main menu: "Data exchange > EPLAN > EXF export".

In the following window, click on "Export > Forms".

Importing forms to COMOS

 The Import/Export dialog is located in the "PlugIns" iconbar. EPLAN forms are created within COMOS as subreports and thus require a plot frame to be created as the master report.

Often the forms are only imported once, since as a rule the templates remain constant for a certain time period.

The Forms tab

- "File"

  A form is selected here. Each form must be imported individually.

- "Plot frames"

  The master report generated in the section titled "Importing plot frames (Page 136)" is specified here.

  A master report has already been prepared within the COMOSDB. When the "EPLAN data exchange" tab is opened, this master report is entered as the default. However, another interactive report template can be set instead by using drag&drop.

Importing forms with identical names

EPLAN forms with identical names can be imported, provided they have different types (extensions).

Their type is appended to their name with an underline character.

Example:

```
EPLAN: ABC.NTS, ABC.NTF

->

COMOS name: ANC_NTS, ABC_NTF
```

6.3.5 Importing project data

6.3.5.1 Exporting projects from EPLAN

The applicable prodcedure is described in the EPLAN documentation. The basic procedure is as follows:

1. Click the "Data exchange > EPLAN -> EXF export" command in the EPLAN main menu.
2. Click "Export > Project" in the following window.

The EPLAN project is exported as an EXF file.

6.3.5.2 General import details

- "Drawing type" (or a "Symbol Type" in the report template)

  EPLAN pages are converted into one of the DETAIL drawing types:

  – Page types "EPLAN page for PLC" and "PLC overview" (COMOS document objects, "EXF settings" tab, "Page type" field) receive the "DETAIL_2" drawing type when they are imported.

  – All other EPLAN pages get the "DETAIL" drawing type when imported.

    Evaluation reports do not have any drawing type.

- Document base object

  All pages receive "@O Documents > EXF EXF import > EXF Standard".

6.3.5.3 The EXF import tab

- "File"

  The EXF file is specified here.

Options group "Templates"

- "Template"

  Here a plot frame that had been converted is stipulated as the Master Report.

  A master report has already been prepared within the COMOSDB. When the "EPLAN data exchange" dialog window is opened, this template is entered as the default. However, another interactive report template can be set instead by using drag&drop.

- "PLC template"

  Here, a plot frame that has been converted is stipulated as the master report, if the "PLC overview" option was activated during the import into COMOS.

  A master report has already been prepared within the COMOSDB. When the "EPLAN data exchange" dialog window is opened, this template is entered as the default. However, another interactive report template can be set instead by using drag&drop.

- "Forms"

  Here you stipulate the document group in which the converted EPLAN forms are located.

  A document group has already been prepared within the COMOSDB. When the "EPLAN data transfer" dialog window is opened, this document group is entered as the default. However, another document group can be set instead by using drag&drop.

- "Start object"

  Optional. Here you stipulate the start object for an article catalog. You should enter the following object here in the COMOSDB:

  "Import > @EXF EPlan > System Special symbols > @A Article/Manufacturer catalog"

Options group "Pages"

- "All / From...to"

  As described above, EPLAN only recognizes single-page documents. These single-page documents are also designated as "pages".

  "All" activated All pages of the project are imported.

  "All" deactivated State in the "From... to" fields which pages are to be imported. In this case an EXF file must already have been stipulated. The page information is taken from this EXF file and offered in a dialog window.

- "Only circuit diagrams"

  Activated: Within this page range only those documents that are classified as "circuit diagrams" are imported. Currently that applies to EPLAN pages with the label "1".

6.3.5.4 The import process

Creating the report

A report with all information on the EPLAN page is produced within COMOS.

Creating engineering objects

When the EPLAN pages are imported, the devices and any other objects placed on them are investigated. The device have a text that designates the accompanying symbol. On the basis of this text, a search for the symbol file is carried out in the COMOS base data, and a search for the symbol base object is carried out in the symbol file.

If the symbol is found, an engineering object is created underneath the report on the basis of this symbol base object.

Allocating article base data and symbols

The device on the EPLAN page can additionally have up to ten different article numbers. The first article number is written in the symbol engineering object generated above, in the first article data field on the "EXF" tab.

This article number is compared with the COMOS base data in the "@A Part-, manufacturer catalog" branch. If a base object with the corresponding article number is found there, this base object is copied under the symbol base object. The symbol engineering object is then no longer based on the variant but instead has the article base object as base object:

```
Import
|- EXF EPlan import/export
   |- DIC_WUPD symbol file
      |- ...
      |- 259 MMT motor with thermistor
         |- 0 MMT motor with thermistor
            |- Siemens
               |- 456abc article1
```

Connecting the report objects

No connection information is stored within EPLAN, instead only the relative position of the objects to one another (under one another, next to one another) constitute the links. The objects are simply connected to one another according to their position.

This function is reproduced in COMOS during the import.

6.3.5.5 Converting individual objects

Allocating a device tag to components

Components without a device tag are collected at a node point. A graphic assignment of these components is performed towards the end of the import: A line is drawn whose height is exactly one grid point is created. Starting from a component without device tag, a search is made to the left for the next device tag.

Object without device tag is element: If a suitable object with a device tag is found, the component without device tag is shifted within the engineering data underneath the object with device tag that had been found. The device tag is taken over.

Object without device tag is device: The device tag is taken over. (In EPLAN it is possible to have device tags that are not unique.)

It is possible that a component cannot be allocated, in which case it remains within the collector nodes.

Terminals

1. EPLAN terminals only have one connector, while COMOS terminals have two. In addition, the two COMOS terminals are classified as "inside" and "outside".

   The second terminal connector is created automatically and connected graphically during the import operation. This is done by drawing a line through the connector that exists and identifies the next connector (in graphic terms) on the document.

2. A terminal strip is not entered in the EXF file for each terminal, but only for specific terminals. The ExF label is not sufficiently unique to include all subsequent terminals of this terminal strip (up to the next terminal strip).

   For that reason the terminals are assigned to the terminal strip as follows:

   This allocation is taken over if there is a unique allocation of the terminal and terminal strip within the ExF data.

   All other terminals are allocated on a purely graphical basis. This is done by positioning the terminals on the document as stipulated within the ExF data, after which the nearest terminal with a terminal strip allocation is searched for horizontally to the left. (This is the same method as in the allocation described above for the device tag.)

Contactors / coils: NClosed contacts, open contacts and changeover contacts

The names of open contacts, closed contacts, etc., are not unique in the ExF data. For that reason the device tag is also accessed. Normally open contacts are created together with a device tag. If a coil with the same device tag is found during the import operation, the normally open contacts are shifted to under the coil (the coil becomes the owner of the normally open contacts).

Path texts

The path texts are taken over and created as report objects. The path designation is written in the description for all objects of the path.

Screening / shielding

EPLAN cables can have several forms of shieldings. The location of the shield connection can be moved.

This function is implemented in COMOS.

Connector points not joined on a connection

In EPLAN connectors can exist on a connection without actually being joined. This is not possible in COMOS, since all connectors are automatically linked to connections. The conversion still remains open concerning this point.

Connections

In EPLAN connectors are automatically joined if they are to be uniquely assigned in a graphical form (e.g. vertically under one another without any obstacles in between).

If you want to suppress an automatic connection of this type, you must place an "interrupter" between the relevant connectors of the two objects.

If a connection has to be "bent", then "diversion points" must be placed, these being objects that only go to one connection and continue on at a right angle.

All these functionalities are reproduced when importing and exporting.

PE connectors

PE connectors are not uniquely marked as such in the ExF data. The conversion still remains open concerning this point.

6.4 Working in COMOS with (imported) EXF data

6.4.1 Prepared labelling systems and objects

If you right-click on the project (working layer) on the "Units" or "Locations" tab, a context menu appears with various objects.

A mixed system is available on the "Units" tab: subunits and also locations can be created underneath units.

Only locations can be created on the "Locations" tab.

This object structure directly implements the "Page numbering" function from EPLAN and may not be changed. See also the section titled "Preparing a COMOS engineering project (Page 127)", subsection "The EXF tab".

Note

If you export to EPLAN, the existing object structure, in connection with the "Page numbering" project option, has a decisive effect on the page numbering created within EPLAN.

In other words: If you plan freely within COMOS and then transport this data to EPLAN, you must ensure yourself that the existing structure of the engineering objects matches up with the "Page numbering" setting in the project.

The labeling systems in the context menu under "New" are based on the base objects in "Import > @EXF EPlan import/export > @EX objects".

This catalog is a copy of the COMOS EE catalog but without its links and references. This means that this catalog always has to be maintained separately.

6.4.2 Prepared objects in the categories

"Categories" are offered underneath the locations. This is purely a COMOS technique. Categories are automatically folders:

- Only specific objects are offered in the context menu of a category object. Thus only specific objects can be created.
- And vice versa: the objects belonging to a category are automatically moved there if they are created at another point.

6.4.2.1 Documents category

Circuit diagram

The circuit diagram has a document-specific symbol bar with the most important symbols.

PLC cross-references list

That is a PLC overview. PLC report templates also have the scripts required for PLC terminals. See also the section titled "Control category (Page 144)", keyword "PLC device box".

6.4.2.2 Terminal strips category

Terminal strips have two levels in COMOS: The terminals are elements of the terminal strip. Before the terminals can be placed, first of all a terminal strip must be created.

Procedure:

- Create a terminal strip (special symbol -91)
- Create one of the offered terminals
- Placing terminals

6.4.2.3 Control category

PLC device box

The "PLC device box" also belongs to the special symbols. This is displayed as a "segment" on the circuit diagram and on the PLC cross-references list report. The "PLC device box" object offers the following elements in the context menu:

- PLC terminal

  PLC terminals can be placed both on circuit diagrams and on PLC cross-reference lists. The PLC terminal can be regarded as a type of channel. PLC terminals are also called "end terminals that can have cross-references".

- "Terminal setting" tab:
  - The channel address is input in the "PLC address" field.
  - The channel type is input in the "Connector type" field.
- Terminal
  - "Normal" terminals possess no address, but can be allocated to a type of channel. The type of channel can be stipulated on the "System" tab.
  - "Normal" terminals can only be placed on circuit diagrams, not on a PLC cross-references list.
- Plugs

  As for terminals.

6.4.2.4 Cables category

Cables as per VDE and the special symbol "-93 cable" are offered here.

Special symbol "-93 cable"

Cables have two levels in COMOS: The wires and screening are elements of the cables. The cable is placed on the report, not the wires (differing from a two-stage terminal strip!).

Procedure:

- Create a cable (special symbol -93)
- Create the wires or screening
- Placing the cable on the report

Alternative:

If only the shields need to be displayed graphically or if you only have one single shielded cable, proceed as follows:

- Create a cable (special symbol -93)
- Create the shielding
- Place the shielding on the report

6.4.2.5 Devices category

All the electrical devices are offered here.

Contacts (normally closed, normally open) for coils or switches

The object first has to be selected in the case of contacts that only exist as elements under the object. Example:

- Select a safety coil
- Select a contact for the safety coil
- Place the contact in the report

6.4.3 Cross-references

Objects that have been placed several times in the circuit diagram are displayed with cross-references. Example: an auxiliary contact of a protective motor switch.

If symbols possess a cross-reference of this type, then the display of the cross-reference can be controlled within the context-sensitive mouse menu for the symbol:

- Mouse-click on the symbol
- Context menu "Settings > Component type"
- Select the type of cross-reference

6.5 Exporting from COMOS (COMOS -> ExF)

This section, which discusses exporting from COMOS to EXF, will only be comprehensible if you have already read the section about importing. The following only describes those input fields and procedures that differ from those for import operations.

 The Export dialogue is in the "PlugIns" iconbar.

6.5.1 The EXF export tab

General points regarding export

The "EXF settings" attribute tab is in the properties of the report.

- "Page type":

All pages are exported as "free graphics", with the exception of pages of types "EPLAN page" and "PLC EPLAN page".

The EXF export tab

- "File"

Here you enter the EXF file to be exported to.

- "Start object"
  - Start object is a document: only this document is exported.
  - "Start object" is a unit, location or document group: all the documents located under it are exported. This also includes those documents that only exist under the start object as a reference or link: the document links are traced back to the original and this is exported as well.
  - "Start object" is blank: all the documents of the entire project are exported.

6.5.2 The Symbol export tab

The "Symbol export" tab enables the export of a previously imported symbol library or a symbol library created within COMOS according to the EPLAN pattern.

The EPLAN methodology regarding symbol libraries

See also the chapter titled "Importing EPLAN symbols to COMOS (Page 132)", subsection "COMOS structure of the imported data".

It is necessary to pay special attention to the following points when exporting symbol libraries:

- The "Component type" field must be given a valid setting on the "EXF" tab.
- This field offers a standard table of all possible EPLAN components.

The "Label" must be filled in. The symbol name is input in the "Label"; this is the description by which the symbol is identified within the symbol file in terms of the contents.

Label	Component type
C	Capacitors
D	Delay and storage devices
E	Various (lighting, heating)
F	Protective devices
G	Generators
H	Indicator modules (optical and acoustic)
K	Contactors, relays
L	Inductances
M	Motors
M6	Motor with 6 connectors
N	Amplifier regulators
P	Measuring and test devices
Q	Main power switchgear (power switches, protective switches)
R	Resistances, potentiometers
SL	Open power contacts
S	Switches
T	Transformers
U	Modulators, converters
V	Semiconductors, pipes
W	Changeover contacts
WM	Changeover contacts, middle
WR	Changeover contacts, right
X	Terminals, plugs, sockets
Y	Electrically operated mechanical devices (solenoid valves, brakes)
Z	Termination, filters, rectifiers

- The "Name" must be unique and should be meaningful

  The symbols belonging to the standard as named as follows:

  1st character: Device ID character

  2nd character: Number of connector points

  Example: Fuses

  There is a fuse F1 and a fuse F3 in the symbol file from EPLAN. The ID character for protective devices (fuses) is F. The number of current paths follows: 1 for a single fuse (1 current path) and 3 for the 3x fuse with three current paths

Symbols tab

Note the following regarding the symbol:

- It must be a quad grid.
- The connectors must be located on the grid points

The Export symbols tab

- "Export to file"

  File name under which the symbol library is to be exported.
- "Symbol file to be exported"

  Root node of the symbol library to be exported. This field is set via drag&drop.

RUPLAN (import) 7

7.1 Base objects and configuration

The base objects of the Ruplan import are located in "Import > RUPLAN". The general configuration of Ruplan imports can be conducted via the Ruplan configurator: "Import > RUPLAN > System > 002 Ruplan configuration", "Settings of the Ruplan import" tab. See also the section titled "Ruplan configurator (Page 153)".

7.2 Options and interface

7.2.1 Open the Ruplan import

To open the Ruplan import, proceed as follows:

- In COMOS, click the "Plugins > Automation > Interfaces > Ruplan import" menu command.

  The "Ruplan import" tab will then appear.

7.2.2 Structure of the Ruplan import window

The "Ruplan import" tab is divided into the following areas:

- "Ruplan data"

  The "Ruplan data" settings are made here.

- "COMOS data"

  The "COMOS data" settings are made here.

7.2.3 The "Ruplan data" area

A complete Ruplan import consists of a symbol file, sheets and devices. However, it is not always necessary to import all three categories. If symbols had already been imported and have not been changed since, then you only need to import the sheets and the devices.

Selecting the symbol file

To select the symbol file, proceed as follows:

1. Click the "..." button in the "Symbol file" column.
2. Within the file selection area, navigate to the folder which contains the symbol file and open it.
3. Mark the file and click on the "Open" button.
4. Click the "Read" button to display the symbol preview.

Select the devices file

The device file contains descriptions regarding the engineering objects that are placed on the sheets.

To select the device file, proceed as follows:

1. Click the "..." button in the "Device file" column.
2. Within the file selection area, navigate to the folder which contains the device file and open it.
3. Mark the file and click on the "Open" button.
4. Click the "Read" button to display the device data in preview format.

Select sheet file

To select the sheet file, proceed as follows:

1. Click the "..." button in the "Sheet file" column.
2. Within the file selection area, navigate to the folder which contains the sheet file and open it.
3. Mark the file and click on the "Open" button.
4. Click the "Read" button to display the sheets in preview format.

Translation file

The translation file serves to replace the placeholders of the type #. See also the section titled "Conversions (Page 152)", keyword "Placeholders".

Saving the text marks

The text marks from Ruplan are imported and saved in COMOS: <comos>\ocx\ComosPlugIns\ KennAbbildung _RP .txt. See also the section titled "Conversions (Page 152)", keyword "Text marks".

7.2.4 The "COMOS data" area

The following table provides an overview of the control elements in the "COMOS data" control group.

Control element	Explanation
"Import symbols" button	This button starts the import of the selected symbol file; i.e. COMOS data is only generated or changed once this button has been clicked.
"Import devices" button	This button starts the import of the selected device file; i.e. COMOS data is only generated or changed once this button has been clicked.
"Import sheets" button	This button starts the import of the selected sheet file; i.e. COMOS data is only generated or changed once this button has been clicked.
"Base data under system" option	If the "Base data under system" checkbox is activated, the imported base objects are generated for symbol files underneath the "Import > RUPLAN > System" node. If it is deactivated, they are saved on a project-specific basis, i.e. in the "Import > RUPLAN > SY_Ruplan import" node.
"Report template" button	Select a report template from the document group "Import > Ruplan". The template is copied and the Ruplan template of the imported symbol file is imported into this copy. The via the button selected report template is displayed in the underneath located text field for control purposes. Prior to importing the sheets, you can define in the COMOS data area whether all sheets or only sheets of a certain area are to be imported (from: Start sheet, to: End sheet).
"Cancel" button	Cancels the import process.
"Open log data" button	Opens the automatically generated log file.
"Target object" field	Via this field a concrete target object can be selected, underneath which the import is conducted.
"Owner"	The "Owner" option is only evaluated if no target object is set. In this case a unit or location is created directly underneath the project, and the import is conducted underneath. The base object on which the unit or the location is based is defined in the "Ruplan configurator".

7.3 Information on importing

Import the symbols and devices into the base project. Afterwards you can import the sheets into the engineering project. You must ensure that the symbols and devices that the imported sheets use are already available as base objects in COMOS. If this is not the case they are not imported correctly.

But you can "reload" base data into the engineering project. If the import data also contains base data, then these are automatically created in the base project, even if the current project is an engineering project: The base project is opened in the background and the base data is entered there.

Currently the data is sorted into folders during the import. This applies to the "Owner" option as well as for the "Target object" option.

Documents are sorted and arranged based on their sheet designations (AP_numbers). All associated sheets and objects are created underneath a folder. The sheets and the COMOS objects with the device name are stored in parallel.

7.4 Conversions

Text marks

The text marks in the Ruplan files are read in and compared with the conversion table stored in the following file: `<comos>\ocx\ComosPlugIns\KennAbbildung_RP.txt`.

In this ASCII file, you can define what is intended to happen with the texts in COMOS.

If the Ruplan file contains text marks that are not yet part of this table and you click the "Save text marks" button, then these missing test marks are entered at the end of the file as "UU" (=unknown). The user has to manually edit the file and has to define how these text marks are supposed to be processed.

There are different options on how to further process the texts. Most of the time you will have to assign a text to an attribute. Another option is to detect, by means of the text, if an object is supposed to be created as a unit or a location.

Placeholders

If the import file contains placeholders for texts, it is checked in the translation file during the import on how to replace these placeholders.

Background: This is not a translation into a foreign language. This file is used to replace placeholders of type #. In Ruplan a placeholder is initiated by means of a # (hash or pound sign). As a rule, placeholders of this type serve as variables, for example to insert a date. The placeholders are replaced by the texts within the translation files in the course of the import operation.

Object conversions

- Ruplan wiring symbols are replaced by dynamic connectors. Background: In Ruplan the intersection and contact points of electrical cables are displayed by symbols of their own. However, in COMOS neither electrical connections nor the intersection and contact points of electrical connections have symbols of their own.
- Ruplan cross-reference symbols are imported as objects and are created on the report as text and graphics on layer 101. The usual connections and links are created within COMOS and can also be displayed or hidden.
- Polygon lines are imported.
- The attributes from the device file are written to the engineering objects as attribute values.

- The unit name from Ruplan is stored in COMOS as the unit label.
- The document attributes (hence the attribute of the plot frame) are taken over and are displayed on the document.

7.5 Ruplan configurator

The general configuration for the Ruplan imports (i.e. the default settings for all Ruplan imports) is conducted on the base data side via the following base object:

"Import > RUPLAN > System > 002 Ruplan configuration", "Settings of the Ruplan import" tab.

"Settings of the Ruplan import" tab

The following is an overview of the attributes of the "Settings of the Ruplan import" tab:

Edit field	Explanation
Base parameter	
"RUPSYS001 Reference object for unit"	If no owner was selected for the concrete import: The base project for the unit which is created under the project node on the engineering side and under which it is imported. If an owner was set for the import: The base object for the API folder, which is generated underneath the owner and under which is imported.
"RUPSYS002 Reference object for location"	Base object for the location or the API folder in the location world; analog toRUPSYS001" | "Reference object for the unit
"RUPSYS003 Reference object for report template"	During the base data import: an empty report template in which the template (the frame) is imported, and is used to import the sheets later.
Individual parameters	
"RUPSYS004 Reference object for dummy document group"	The document group in which the templates and the sheets are imported to.
"RUPSYSAK01 Label for wire"	Default label which is used if an imported wire does not have a label.
"RUPSYSHK01 Identifier for main objects"	Text mark which defines that an object is imported as a main object.
"RUPSYSKST01 Identification for separation bar"	Text mark which defines that an object is imported as a separation bar.
"RUPSYSBR01 Bridge type"	All imported bridges get this type.
"RUPSYSPOTZIEL Identification for potential cross-objective"	Text mark which defines that an object is imported as potential cross-objective.
"RUPSYSPOTREF Identification for potential references"	Text mark which defines that an object is imported as a potential reference.

Edit field	Explanation
"RUPSYSDokGrp Identification for AP number"	Text mark which defines that an AP number will follow.
"RUPSYSBMK Identification for BMK"	Text mark which defines that a BMK follows.
"RUPSYSANL Identification for unit"	Text mark which defines that a unit follows.
"RYUPSYSORT Identification for location"	Text mark which defines that a location follows.
"RUPSYSCLASS Set symbol class"	Activated: If base objects are not classified (i.e. were not imported with the "Device" class): During the import of the device data you can search for the entered identification key in RUPSYSHK01"|" Identification key Base objects. For all devices that have this identification key: The class of the base object is set to "Device".
"RUPSYSLOGPOT Importing logical potentials"	Self-explanatory.
"RUPSYSKNR Identification for channel number"	Text mark which defines that a channel number follows.
"RUPSYSCON Identification for connection"	Text mark which defines that a connection key follows.
"RUPSYSPOTART Identification for potential type"	Text mark which defines that a potential mode follows.
"RUPSYSPOTTEIL Identification for potential part"	Text mark which defines that a potential part follows.
"RUPSYSUNITLOCATION Creating units and locations"	The units and locations are also imported from Ruplan. Thereby the in RUPSYS001| Reference object for the units, or RUPSYS002| Reference object for the locations are used.
Parameter for engineering objects	
"RUPSYSKab01 Reference object for cable"	Base object for cables.
"RUPSYSAder01 Reference object for wire"	Base object for wires.
"RUPSYSBI01 Reference object for document reference 1 (ext.)"	Base object for a reference to another project.
"RUPSYSBI02 Reference object for document reference 2"	Likewise.
"RUPSYSBI03 Reference object for document reference 3"	Likewise.
"RUPSYSCONINFO Connection information"	Base object for connection information.
"RUPSYSANS01 Reference object for connector reference 1"	Base object for a connection reference.
"RUPSYSANS02 Reference object for connector reference 2"	Likewise.
"RUPSYSANS03 Reference object for connector reference 3"	Likewise.

PLANEDS (import) 8

8.1 Aim

You can use the "Planeds import" import interface to import Planeds data from an Access database (*.MDB).

The following Planeds data can be imported into the base project or into COMOS engineering projects:

- Base project
  - Function code
  - Device types
  - Wire types
- Engineering project
  - Devices
  - Pltplaces
  - Function, location tree assignment
  - Connections
  - Cables
  - DXF documents
  - Graphical connections

8.2 Preparing import data

Procedure

To import Planeds data into COMOS, you must perform the following steps:

1. Export your data in CSV format in Planeds.
2. Import the CSV files into an Access database (*.MDB).
3. Select the Access database for the import in COMOS.

Note

The general structure of the Planeds interface is described here. You may need to customize it, depending on your particular project.

Base objects

The base objects for the Planeds import are located in the "Import > @Planeds" node.

8.3 Options and user interface

8.3.1 Opening the Planeds import

To open the Planeds import, proceed as follows:

- Double-click on the "Import > @Planeds > IMPORT Planeds import" base object.

  The "IMPORT Planeds import" tab will then appear.

8.3.2 Structure of the Planeds import window

The following tabs are available in the "IMPORT Planeds import" window:

Tab	Target of the import
"Function code"	Structure for the base data.
"Device types"	Base data with device numbers.
"Wire types"	Base data for cables and wires.
"Devices"	Objects for the location page of the engineering project.
"Pltplaces"	Objects for the unit page of the engineering project.
"FW-OW"	Links of objects on the unit and location page.
"Connections"	Connection information of objects.
"Cables"	Cable and wire information for engineering objects.
"Documents"	Conversion of DXF documents into COMOS documents.
"System"	The Planeds objects are assigned to COMOS base objects in the unit and location trees here; terminals are assigned too. These settings must be made prior to the import.
"Graphical connections"	Connection lines from DXF documents are replaced by COMOS connection lines.

Importing manufacturer catalogs or manufacturer devices

9

9.1 Aim

Aim

When selecting manufacturer devices, a request that was originally used to perform engineering is replaced by a product. The request has certain attribute values, which represent templates for the to be searched product. You manually create a data set in COMOS for each product.

You have two options for connecting manufacturer catalogs:

- Transfer of only product data (symbols, attributes) from manufacturer catalogs
- Base data import from manufacturer catalogs

9.2 Basic principle of importing

Basic technique

The base data has the following structure:

- There is a central branch for each technical area.
- At the top levels of the branch, the following applies to the base objects:

  The "Request" option is activated.

  These base objects are referred to as "requirement objects".
- At the lower levels, the following applies to the base objects:

  The "Request" option is deactivated.

  These base objects are referred to as "manufacturer devices".

9.3 "Product data" function right

Introduction

You must have the corresponding rights in order to prepare orders. If you are an administrator you can allocate the necessary rights to the user via the rights management.

Procedure

To assign the "Product data" function right, proceed as follows:

1. Mark your project in the Navigator and press the <Ctrl+A> keys on the keyboard.
2. Click on the "Rights" tab in the window that opens:
3. Activate the "Function rights" option.
4. Right-click on an employee or employee group in the lower area of the window.
5. Click on "Properties" and select the name of the employee or employee group.

   A window is opened.
6. Activate the "Product data" option in the window.
7. Click "OK" to confirm your selection.

You find additional information on this topic in the "Rights" manual, keyword "Rights administration".

9.4 Importing a catalog using a read processor

9.4.1 Parser

Catalog format

When a manufacturer supplies a product data catalog, the catalog may have one of the following formats:

- ASCII
- EXCEL
- XML
- MDB

To use this catalog in COMOS, you need to convert the data into a comparable standard format.

A read processor (parser) is written for each of these catalogs which reads in the data contained in the file and supplies it in a format that has been standardized for COMOS.

Base object

The parsers are found in the COMOS base objects, in the "@01 Material > EIC Electrical, measuring and control engineering > 04 Manufacturer catalogs" node.

The data is first distinguished according to type, and then according to the manufacturers.

Structure of parsers

The parsers contain attributes which have the relevant administration information for the corresponding data type, e.g., path to catalog file, coding. Parsers have a query, the illustration matrix, as an element. The structure of the illustration matrix is based on requirements.

Example

Company A supplies a file with terminals. The parser reads this file in and shows the full quantity of terminals. Special parsers, which inherit this quantity, are created underneath the general parser. The inherited quantity is then filtered according to certain criteria, e.g. N or PE terminals.

New format

If a new format is to be supported, another parser will need to be created.

9.4.2 Illustration matrix

Query

Once you have performed the mapping, filtering, and sorting procedures, you can select the to be used quantity of product data.

You can make this selection in two ways:

- Create the selection in the form of COMOS objects in the base data. An additional window opens in which you select the node under which the COMOS objects are to be created.
- Write the selection to the corresponding processor (owner of the query) in XML format. The processor is then available for supplying data in response to a request.

9.4.3 Editing the illustration matrix

Overview

During the initial read-in, all identifiable fields are mapped in a query.

Procedure

To edit the illustration matrix of a parser, proceed as follows:

1. Open the base project.
2. Select the "Base objects" tab in the Navigator.

3. Select "@01 Material > EIC Electrical engineering > 05 Manufacturer catalogs", along with the required manufacturer catalog.
4. Open the illustration matrix of the catalog.
5. Right-click on the required column and select the "Mapping" command in the context menu.

   The "Map source names on COMOS attributes" window opens.
6. If you want to assign a COMOS attribute to a field, enter the nested name of the COMOS attribute in the "COMOS attribute name" column.
7. Deactivate the option in the "Active" column for the fields you want to hide.
8. Define the required units by mapping the unit from the file to a COMOS unit.

   The values are converted accordingly for display purposes, and are then saved.
9. Create more columns if necessary.

   Newly created columns are filled in via a script.
10. If languages are available, they are taken into account in the query in the form of additional columns.

    This is not done via the parser.
11. Click "OK".

Result

Your settings are saved. You return to the illustration matrix.

You find additional information on this topic in the "Queries" manual.

9.4.4 Selecting to be used product data

Overview

Following mapping and any filtering and sorting, you select the quantity of to be used product data.

Procedure

To select the quantity of product data to be used, proceed as follows:

1. Select the required product data from the illustration matrix list.
2. Right-click on the header of the column in which you have selected the product data.
3. Select the "General XML for selection" command in the context menu.

Result

The selection has been written to the relevant processor in the form of an XML file. The processor is then available for supplying data in response to a request.

9.4.5 "HSD Ordering data" tab

Overview

You define which catalogs are valid for a request via the "Available catalogs" control group found on the "HSD Ordering data" tab. Any number of link attributes can exist in the control group. The processors used to transfer data or the relevant base object nodes for selecting products from COMOS objects are set in the link attributes.

If these attributes are missing, the familiar "downward compatibility" function applies automatically.

9.5 Selecting a manufacturer device base object for an engineering object

Procedure

To call a manufacturer catalog on an engineering object, proceed as follows:

1. Open the properties of the object whose manufacturer catalog you want to call.
2. Select the "General" tab.
3. Click the "Set pointer ..." button next to the "Base object" field.

   The "Look up Manufacturer Device for" window opens. All activated manufacturer catalogs of the base object are displayed.

   The product-relevant attributes are displayed in the upper area, along with the request value and the "Selection" list for the device value. The quantity of available manufacturer devices is decreased by successively restricting the values for the individual attributes. All values which are available on the manufacturer devices are offered for each attribute.

   The lower area contains a list of all manufacturer devices available for selection.
4. Select the required manufacturer device.
5. Click "OK".

Result

The data is transferred to the request.

The following algorithm applies:

Attribute is product-relevant	Values in OwnValues
Yes	Values are written to the OwnValues, if they are empty. Values are also written to the XValue (ProductValue).
No	Values are always written to the OwnValues of the attribute.

9.6 Catalog devices, using the example of FESTO

9.6.1 Overview of catalog device importing

Aim

You can use the manufacturer device search on the report to select manufacturer device symbols from a library. The library must be present as an Access database. There are two applications for this:

- Replacing a symbol in the case of an object placed on the report
- Placing a new symbol on the basis of a library entry

Supported symbol libraries

The manufacturer device search on the report currently supports the following manufacturer device libraries:

- FESTO

9.6.2 Example database for FESTO

You can find an example database in the following location:

COMOS database installation directory; path: "... > SO1 > ManufacturerCatalogs > Festo".

9.6.3 Reference of options and conversion settings

Options

Note: the "Options" dialog contains the default settings every time it opens. The information in the "Options" field is not evaluated in the conversion settings.

- Conversion

  Standard conversion factor. The "Conversion" field is locked in the options. A different conversion factor can be entered in the conversion settings.

- Mirror

  Controls the mirroring of the symbol.

- Alignment

  Controls the alignment of the symbol. Each symbol has a placing position, and the "Alignment" option defines the position of the symbol in relation to the placing position. In this case, the device label and other imported symbol components belong to the symbol.

- Device label

  Controls where on the symbol the device label is displayed.

  The "Device label" and "XML" options are linked.

- Texts

  Controls the import of the symbol texts.

- Connectors

  Controls whether the graphic of the connectors is shown.

- XML

  The coordinates of the device label are taken from the imported XML file of the symbol.

  The "Device label" and "XML" options are linked.

- Action

Settings

- "Manufacturer"

  In this field, only the entry "FESTO" is currently supported.

- "Manufacturer device", "Description", "Type", "Symbol", "Classification"

  In these fields, a search term for the column of the same name can be entered in the list below in each case.

- "Conversion"

  Symbols in the FESTO library are larger than the standard symbols in the COMOSDB.

  Enter a numerical value in the "Conversion" field:

  – Value less than 1: The symbol's size is reduced during the import.
  – Value equal to 1: The symbol is imported in its original size.
  – Value greater than 1: The symbol's size is increased during the import.

- "Option"

  Contains the settings of the "Options" dialog in short form.

Each change to the options must be confirmed by clicking on the arrow button in the "Display" column.

9.6.4 Assigning a catalog device

Requirement

A "Pneumatics" or "Hydraulics" type document is open.

Assigning a catalog device in the case of a placed object

To assign a catalog device, follow these steps:

- Select a symbol on the report.
- In the context menu, select the command "Select a symbol from the catalog".
- Click the "Open" button.

  Open the FESTO library. All the library entries are listed in the bottom area.
- If necessary, you can narrow down the entries that are listed. See also chapter Narrowing down which catalog devices are available (Page 165).

  In the COMOS standard model, the "Manufacturer device" value is transferred to the " HSD.M001 Article number" attribute. If the symbol selected on the report has an article number, this value is automatically entered in the "Manufacturer device" field.
- Define the scale factor. See Reference of options and conversion settings (Page 163).
- Select a symbol from the list in the bottom area.

  Confirm with "OK".

9.6.5 Importing a catalog device

Requirement

A "Pneumatics" or "Hydraulics" type document is open.

Importing a catalog device

To import a catalog device, follow these steps:

- Click in the empty area of the report.
- In the context menu, select the command "Select a symbol from the catalog".
- Click the "Open" button.

  Open the FESTO library. All the library entries are listed in the bottom area.
- If necessary, you can narrow down the entries that are listed. See also chapter Narrowing down which catalog devices are available (Page 165).
- Define the scale factor. See Reference of options and conversion settings (Page 163).
- Move a symbol to the report using drag&drop.

  A symbol preview is displayed during the drag&drop operation.
- Confirm with "OK".

  The symbol and the device label are displayed.

9.6.6 Narrowing down which catalog devices are available

Requirement

The "Import manufacturer device symbol" dialog and the FESTO database are open.

Using a fixed search value

To narrow down the group of symbols available to a single hit, proceed as follows:

- Enter a value in a field that has a check mark in the "Display" column.

  Example: In the "Manufacturer device" field, enter the value "1002502".
- Click the arrow button in the "Display" column.

Only one entry is displayed in the list of available symbols.

Imprecise search

To narrow down the group of symbols available to a smaller number of hits, proceed as follows:

- Enter a value plus a wildcard in a field that has a check mark in the "Display" column. The wildcard is the "*" character (asterisk).

  Example: "99*" (without quotation marks) will find all values beginning with "99".
- Click the arrow button in the "Display" column.

Only entries with appropriate partial values are displayed in the list of available symbols.

Fixed fields

The following fields may not be used for narrowing down the list:

- Manufacturer

  This field must contain the value "FESTO", as only the FESTO library is currently supported.

- "With SymbolXML only"

  This field must contain the value "True", as only the import of XML symbols is currently supported. The XML symbols are converted into VBScript during the import process. In COMOS, the symbols are described with the aid of VBScript.

9.6.7 Storage in the base data

FESTO catalog

The base data for the FESTO catalog devices is stored in the following location:

"01 Material > FDS > FESTO manufacturer catalog"

9.7 ECAD components import

Overview

You have the option of importing partial stocks from the ECAD catalog into new or existing COMOS objects. You can also use the import process to update data that has already been imported.

Preparing the import

The import is based on the control table located in the base project ("@10 > EIC > SYS > Import") which you open in the menu bar using the "Administrator > Base data > Standard tables" command.

The attributes from ECAD are assigned to the COMOS attributes in this standard table. You do not need to assign the ECAD and COMOS attributes 1:1.

The standard table is made up as follows:

Column	Description
"Name" column	Each row in the standard table requires a unique name, which you are free to choose. With the default setting, the name of the attribute is entered here.
"Description" column	This column displays the description of the COMOS attribute. This column is not updated automatically. If you change the descriptions of the attributes, the new descriptions must be entered here manually. The import will still work even if there are incorrect entries in this column.
"Value1" column	Full name of the COMOS attribute used during the import process (including the name of the tab). The COMOS attributes are located in the base project in: "@10 > BAS".
"Value2" column	Name of the assigned ECAD attribute.
Subsequent columns	Other imports that have no significance for ECAD.

You can adapt this list to suit your requirements.

Preconfigured import

When text files are created in ECAD format, each line of text has an initial marker ("tag"). The following tags are recognized and imported:

Tag	Description
NO	Main components
TD	Mechanical data
ZB	Accessories Can be controlled using the "Load accessory" and "Log all" options.
DD	Attributes under manufacturer on the "Order data" tab. K1 data sets (channels) are also imported. The following channel types are taken into account: COIL, MAIN, AUX, PRIM, SEC, UNI and PCL.

Conducting the import

To open the window for the import process, select the following command in the menu bar: "Administrator > Base data > Open ECAD components import".

The following table describes the control elements of the ECAD components import:

Control element	Description
"File" field	This field shows the VRG file that you wish to import.
"Select importing file" button	Use this button to open the "Open" window. Select the required file. Confirm your selection with the "Open" button.
"Target" field	This field shows the node under which the data is to be created. Drag the required note from the Navigator to this field using drag&drop. You must import one block of manufacturer devices at a time underneath the relevant branch for the request. New objects are created under the target node. The attributes are updated in the case of existing objects.
"Load accessory" option	Lines with the ZB tag are imported and created underneath the main components (NO tag). Accessories are those devices that are available for a specific other device but are not managed individually. Accessories are, therefore, used as an "element" on the "Elements" tab of another device from the selection catalogs. The main reason for adding accessories to another device as an "element" is that you can create complete order lists automatically in this way.
"Log all" option	A log file is created during the import process. This file is located in the same directory as the file to be imported. The file name of the log file is made up of the name of the file to be imported, a consecutive number, and the file extension ".pcl". The log file is always newly created. • Off: Only errors are logged. • On: The entire import process is logged (for example, which objects were created or which information was written to where).
"ASCII-ANSI character set conversion" option	An ASCII character set is converted to ANSI by means of the Windows routine.

The import process is displayed by a progress bar.

Information on earlier interfaces 10

10.1 Process-neutral interface (PNI)

Help on the process-neutral interface

The process-neutral interface is out of date, but continues to be offered in COMOS for compatibility reasons. You can obtain a help document for the PNI from customer support by citing the document number HB81D04.

10.2 PCS 7 - COMOS data transfer via COM interface

Help on COM/IEA interface for PCS 7

This interface is out of date, but continues to be offered in COMOS for compatibility reasons. You can obtain a help document for the COM/IEA interface for PCS 7 from customer support. The interface is described in the help documents for COMOS 9.1.