SIEMENS

SIMATIC

Process Control System PCS 7 CFC for SIMATIC S7 (V9.0 SP4)

Function Manual

04/2020
A5E41356073-AE

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

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

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
Division Process Industries and Drives
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E41356073-AE
Ⓟ 04/2020 Subject to change

Table of contents

1 Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit
https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under
https://www.siemens.com/industrialsecurity.

2 What's new in CFC?

CFC V9.0 SP4

Compared with the previous version V9.0 SP3, version V9.0 SP4 includes the following enhancements or changes:

- You can now use a single tab "Properties - Block/Chart" dialog box, "Type update settings" tab (Page 442)" to set/reset "Pink" parameters and other attributes of a block parameter to be ignored from type update. The "Ignore for type update" option has been removed from the "Properties - Input/Output" dialog box.

CFC V9.0 SP3

Compared with the previous version V9.0 SP2, version V9.0 SP3 includes the following enhancements or changes:

- New field for signal assignment in the Technological Editor:
  In the Technological Editor, a new "Assignment" field added for the "Signal" attribute of control module parameter/signal (CV).
  The input of a signal processing block (channel block) will be assigned to the signal of the technological block. With this signal assignment, the signal processing block is determined if there is no direct connection between the signal processing block and the technological block.
  You can find additional information on this in the section "Attributes and attribute values of a control module (Page 214)" and in the section "Variant 4" in "Signal processing for HW signals (IO tags) of control modules (Page 234)".
- New "Type update settings" tab:
  A new "Type update settings" tab is added in the "Properties - Block/Chart" dialog box. In this tab, you can select from various attributes of a block I/O to exclude them from the plant type synchronization (type update).
  You can find additional information on this in the section ""Properties - Block/Chart" dialog box, "Type update settings" tab (Page 442)".

CFC V9.0 SP2

Compared with the previous version V9.0 SP1, version V9.0 SP2 includes the following enhancements or changes:

- New "Avoid implicit dead times" button
  In a CFC, if a feedback in the new interconnection is detected then an automatic correction in the sequence of block execution is carried out to avoid this feedback. This automatic correction results in dead times.
  This automatic correction can be switched on/off with the new button provided to avoid implicit dead times. The default setting of this button is off.
  You can find additional information on this in the section "Display of feedback in the signal processing (dead time) (Page 177)".
- New "Ignore for type update" option
  A new option has been added to ignore the parameter value for the type update when synchronizing the corresponding template.
  You can find additional information on this in the section ""Properties - Input/Output" dialog box (Page 450)".

CFC V9.0 SP1

Compared with the previous version V9.0, version V9.0 SP1 includes the following enhancements or changes:

- Reporting
  The logging functions during downloading have been extended.
  In the "Reports" dialog box, the "Archive" button allows you to open a dialog for selecting and displaying log files with detailed information about the download.
  Information on the download can now also be stored in XML format in addition to the existing log files. These XML files with the information on the downloaded changes can be compared in the VXM if needed.
  You can find additional information on this in the section "Change log and ES log (Page 401)".
- New attributes on technological objects
  With technological objects, such as control modules, the "Low scale value" and "High scale value" have been added. The attributes can be edited at the type of technological object and are only relevant for analog signals (input and output signals).
  You can find additional information about this, for example, in the section "Attributes and attribute values of a control module (Page 214)".

CFC V9.0

Compared with the previous version V9.0, version V8.2 SP1 includes the following enhancements or changes:

- Technological configuration:
  For technological configuration, the Equipment Phase "EPH" is provided to enable control of multiple lower-level equipment modules. The lower-level equipment modules (EM) are no longer permanently assigned in this case, but instead requested at runtime, allocated and then released again.
  The equipment phase of the assignment partner is at the process control end when interfacing to SIMATIC BATCH.
  You can find additional information on this in the section "Basics of the equipment phase (Page 299)".

CFC V8.2 SP1

Compared with the previous version V8.2, version V8.2 SP1 includes the following enhancements or changes:

- New attributes
  - For the "Message" object of control modules, the attributes "Single acknowledgment", "With acknowledgment", "Trigger action", "OS area" and "Batch ID" have been added.
  - The attribute "Version" has been added for control modules and equipment modules.

  The new attributes participate in the data exchange between PCS 7 and COMOS.
  You can find additional information on this in the section "Attributes and attribute values of a control module (Page 214)".
- Feedback in the signal processing of an interconnection
  A delay time in the signal processing of an interconnection is detected and displayed at the interconnection in the CFC.
  You can find additional information on this in the section "Layout of interconnections (Page 76)".
- Data exchange with COMOS via XML format
  On export to an XML file, selected technological objects can now be exported using the new menu command "Export selected objects to XML".
  You can find additional information on this in the section "Data exchange with COMOS via XML format (Page 336)".

CFC V8.2

Compared with the previous version V8.1 SP1, version V8.2 includes the following enhancements or changes:

- Chart-based runtime group management for blocks of CFCs
  For a CFC with activated chart-based runtime group management, the blocks are automatically managed in separate runtime groups based on the chart, i.e. the blocks of the CFC are organized in runtime units that are permanently assigned to this CFC.
  When F-blocks are also used in this CFC, not only is a runtime group for blocks automatically created in the "standard program", but a runtime group for the included F-blocks as well.
  You can find additional information on this in the section "Chart-based runtime group management for blocks of CFCs (Page 192)".
- "Type update in RUN" for default value changes in the interface of blocks / block types
  In the case of default value changes on the interface of a block/block type, this block can be downloaded with the type update in RUN. It is no longer necessary to put the CPU in STOP mode for this.
  You can find additional information on this in the section "Type update with a CPU 410-5H PA (Page 113)".
- The prompt for the password of the safety program for CFCs with fail-safe blocks has been changed.
  You can find additional information on this in the section "Special features of F-blocks in CFCs (Page 102)".

See also

Selective download of individual charts (Page 378)

Operator control and monitoring (Page 170)

Overview of data exchange with COMOS (Page 332)

References at the parameters of a control variable (Page 244)

Configuring enumerations (Page 164)

Attributes of the objects of an equipment module (Page 291)

3 Introduction

What is CFC?

CFC (Continuous Function Chart) is an editor with a graphical user interface, an extension based on the STEP 7 software package. It is used to create the entire software structure of the CPU and uses pre-configured blocks. The editor lets you insert blocks into function charts, assign block parameters and interconnect blocks.

Interconnecting means that values can be transferred from one output to one or more inputs during communication between the blocks or other objects.

Operating principle

You work with graphic tools in the CFC editor: You drag-and-drop pre-configured blocks from the pool of blocks to your chart. The chart is a kind of "drawing board". You interconnect the block inputs and outputs via a mouse click. There is no need to consider details such as algorithms or the assignment of machine resources, in other words, you can fully concentrate on the technological aspects of your configuration.

The runtime properties of blocks are assigned default settings you can adapt individually. Since individual blocks or groups of blocks can be copied or moved from chart to chart, you can save a considerable amount of time. Interconnections between the blocks are retained.

After having created all functions, generate the executable machine code by means of mouse click, download the code to the target system and debug it using the CFC debugger functions provided for this purpose.

Target systems

CFC can be used to configure various target systems. Operations performed by the user when working with CFC are identical to a great extent.

It is possible that parts of this help do not apply to the target systems that differ from SIMATIC. This affects the description of the STEP7 and PCS 7-specific blocks in particular. In this case, please read the Help for the superposed applications.

Note

MW0 cannot be used for operation with an S7-300 module, because it would be overwritten each time a block with startup characteristics is called, for example, FC 70 (RESTART).

Pool of blocks

You can import blocks required in CFC from block libraries, the master data library or other projects, or you can create your own blocks.

Note

CFC is "upwards compatible". Modification of programs using tools other than those of the CFC editor will lead to inconsistencies.

Compatibility with earlier projects

With the current CFC version you can read and display data created using an earlier CFC version. You can edit these data. At the first write access, these data are converted (following a system prompt) in order to ensure compatibility. However, the version is not "downward compatible". You cannot use the V5.x CFC editor to edit data created or converted under version 6.0 or later.

Additional information

You can find brief instructions relating to the configuration steps required to create a project and download it to the AS in the section: Getting started (Page 19)

4 Getting started

Overview

This section explains the procedures for configuring your S7 target system:

1. **Creating the project structure**
   In the SIMATIC Manager , you need to create a chart folder for the CFC below the hierarchy level of the program folder.
   Create the specific CFC charts in the chart folder, either in the SIMATIC Manager or directly in the CFC editor.
   In PCS 7, you use the "New Project" wizard for these steps.
2. **Creating blocks (optional)**
   CFC works with ready-made blocks. These can be blocks from libraries, other programs, or block types created by you. You can create block types in CFC by compiling the charts.
3. **Inserting and importing blocks**
   Block types required for the project must be implemented and, in some cases, imported to the project according to the target system used.
   You insert blocks into the chart from the catalog using drag-and-drop or copy a block from another position to the S7 program block folder. Inserting a block in a chart creates a block instance with a name that is unique throughout the chart. You can create any number of block instances from each block type. When a block is inserted in a chart, the block type is imported to the CFC data management.
4. **Assigning parameters and interconnecting the blocks**
   You can assign parameters to block I/Os and interconnect them. Targets of the interconnection are other blocks, nested charts or shared addresses. You can specify textual interconnections at block/chart inputs whose interconnection target is not yet in the chart folder. This interconnection remains open until the referenced target of the interconnection is available and the interconnection is made via a menu command.
5. **Adapting the runtime properties**
   The runtime properties of a block determine how the block is included in the processing of the entire structure on the PLC. The properties are decisive for the response of the target system in terms of reaction times, dead times or the stability of time-dependent structures, for example, control loops.
   When it is inserted, each block is assigned default runtime properties. A block is inserted in a task at the position you specify. You can change the position at which the block is installed and other attributes later if necessary.
6. **Compiling the CFCs**
   When a program is compiled, all charts of the active CPU are converted to machine code. Different compilers are used, depending on the target system; the call is identical, however. If you compile as a block type, only the individual chart is compiled.

7. **Downloading the CFC program**
   You can download the CFC program to the CPU after compilation.
8. **Testing the CFC program**
   You can test the compiled and downloaded program. The scope and type of test functions differs depending on the target system. In the test mode, you are connected to the automation system online.

Essentials of CFC 5

5.1 CFC in the STEP 7 environment

STEP 7 components

The following STEP 7 components are important when working with CFC:

- SIMATIC Manager
  The SIMATIC Manager is used as the central database and to coordinate the tools and objects for all target systems. It manages tools and data and is used, for example, for creating and modifying project structures (CPU, CFC charts) and for starting the CFC editor.
- STEP 7 tools
  These tools are used to create the input data for CFC and I/O data which you can reference from the CFC:
  – LAD, STL, SCL
    Programming languages used to create block types for S7. After you have created block types in these languages, you can import them into CFC and configure them.
  – Symbol table
    The process signals and other shared addresses such as bit memories and data blocks are assigned symbolic names and are entered in the symbol table of STEP 7. The CFC can use these names.
- SFC (PCS 7)
  Engineering tool used to create technological sequential control systems. With sequential control systems, you can control functions in basic automation using operating and state changes (these functions are normally configured with CFC).

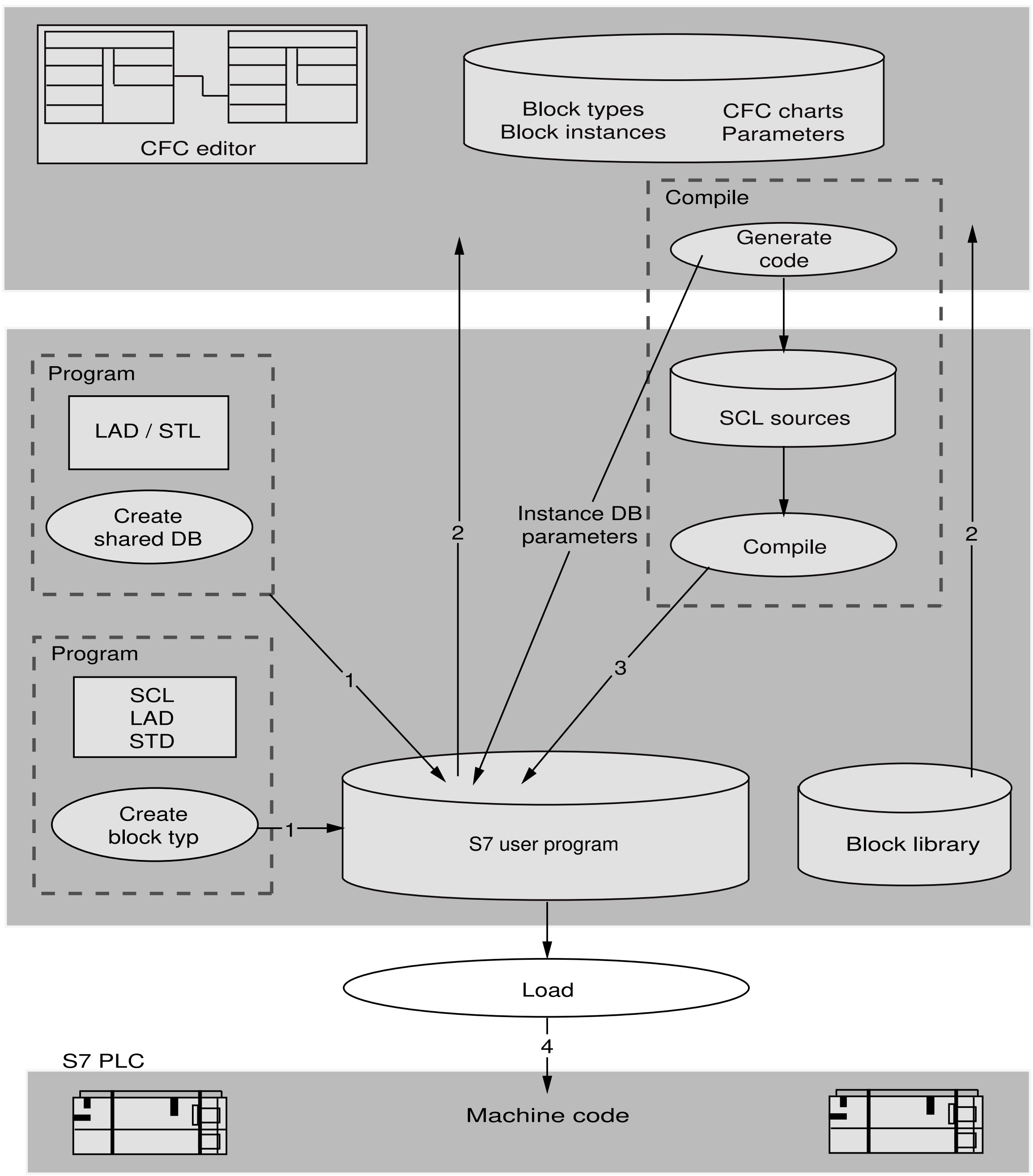
Interaction between CFC and SIMATIC Manager

The following should be observed when working with the SIMATIC Manager :

- You can only delete charts, chart folders and projects using the SIMATIC Manager if no chart in the particular chart folder or project is currently being edited in CFC.
- Projects that contain CFC charts may not be saved on diskettes, neither using the "New project" nor the "Save project as" functions.
- You should only generate chart reference data in the CFC. If you generate reference data in the SIMATIC Manager , you will no longer be able to download changes to this program online.

Flow of configuration data

The figure below shows the flow of configuration data in S7, starting at the creation of block types and going through to the point where the machine code is downloaded to the automation system.



Data flow for configuring an S7-CPU

1. (Optional) The block types are created with STEP 7 tools (for example, with SCL) and saved to the S7 user program.
   (Optional) Global data blocks are created with the LAD/STL editor and stored in the S7 user program.
2. Charts in which blocks (taken from the S7 user program or block libraries) are inserted, configured and interconnected are created in the CFC editor.
3. The CFC charts are compiled. A CFC program is generated.
   Note: In CFC, you can create block types by compiling charts with chart I/Os as block type. Restriction: These charts cannot contain nested charts. Block types created in CFC can also be saved to a library.
4. The CFC program is downloaded to the CPU.

5.2 Blocks in CFC

Functions as blocks

In CFC, you work with preconfigured blocks that execute a specific function. You insert these function blocks in the chart, interconnect and configure them.

Note

The initial value at the output of the CFC blocks is "1", regardless of the values at the inputs. This means the following logic is supplied with "1" as long as the block is not processed.

The block type

A type definition exists for each function block and specifies the following:

- Algorithm
- Type name
- Data interface (input and output parameters)

The type name is an abbreviation or acronym of the function, for example:

- **CTUD** (COUNT UP and DOWN)
  For edge-triggered up/down counters
- **COUNT_P**
  Up/down counter which is triggered at the positive edge of a binary signal (depending on the setting)
- **ADD_R**
  A simple function that adds input values and transfers the result to the output

The type definition also determines the data types of the I/O parameters. In the following, these I/O parameters are referred to as block I/Os, since this is how they appear in the graphical display of the block.

The data type of an input or output specifies the values it can accept.

Examples:

BOOL: Boolean type that can only adopt the values 0 or 1.

STRING: Character string type that can contain a character string as its value.

You can find information about other data types in the section: Data types for S7 (Page 159)

The block instance

When you insert a block in your CFC chart, you generate an instance of this block type. In this context, “instance” refers to an instance of the selected block type.

You can create any number of block instances from a particular block type. You can assign names to these block instances, interconnect and assign parameters to them without changing the functionality specific to the type.

One useful aspect of this type instance concept is, for example, that subsequent local changes to the block type can be automatically distributed to all block instances.

Multiple instance blocks

Functions may consist of various sub-functions. These sub-functions are implemented as blocks and merged in order to create a complex block. This can be a control (internal) block, for example, that contains a message block and an operator control block.

Multiple instance blocks can be created in CFC by interconnecting different blocks (functions) and assigning suitable parameters to those blocks. This chart is subsequently compiled as block type.

Blocks with variable number of inputs

CFC contains blocks whose number of inputs is variable and can be changed in the CFC chart. Such blocks are known as generic blocks. The AND block, for example, is a block with a variable number of inputs.

Block families

Blocks are assembled in groups (block families) according to their functional properties. Each block is assigned a family identifier when it is created. The following blocks, for example, form a family:

- Conversion blocks for adapting various data types
  CONVERT (e.g., BO_BY, BY_DW, W_DW)
- Multiplexer blocks =
  MULTIPLX (e.g., MUX8_R, MUXn_DI)
- Blocks with mathematical (floating point) functions
  MATH_FP (e.g., SQRT, ADD_R)

The block family names are also used, for example, as a sort criterion in the CFC catalog.

Organization blocks

The interface between the CPU operating system and the user program is formed by tasks known under S7 as organization blocks (OBs). These OBs are used to execute specific program sections. For example, there are OBs for:

- CPU startup (cold restart, hot restart)
- Process alarms
- Cyclic interrupts with different time bases

Organization blocks or tasks do not represent blocks in the context of CFC; since they can neither be inserted nor edited in CFC. In CFC, the OBs are displayed in a run sequence after calling the runtime editor.

Additional differences

Blocks also differ in their type. When a block is created, it must be declared as a function block (FB), function (FC) or basic operation (BOP).

- The FB has a memory function. Existing data can be accessed across several processing cycles. Data access is enabled by creating a data block (DB) for each block instance. In a complex block, however, the FB contains additional subsidiary FBs, but only one common DB.
- The FC does not have a memory function. The values generated by the block are processed directly. No data block is required here. Outputs of an FC are not assigned default values.
- The BOP does not have a memory function, the same as the FC. Its basic operations form part of the CFC program and are entered as SCL statements during compilation and used for basic functions such as AND and OR.

Note

In case of a warm restart of the CPU, the BOP inputs receive the last value of the pool DB.

Special situation: Overlapping blocks

Overlapping blocks are inserted or moved to a chart where they have insufficient space. In this case, they overlap other objects completely or partly.

The overlapping blocks are displayed in another color and without the visible block I/Os. The color can be set in the "Color settings" dialog. Their interconnections and entries in the sheet bar are also hidden, but nevertheless exist. After positioning a block on a free area in the chart, the overlapping blocks are displayed as normal again with their previous interconnections.

5.3 Automatic naming

Introduction

Automatic naming in this context refers to objects generated, copied or moved via SIMATIC Manager and/or the CFC/runtime editor.

Creating

Each object in a project is automatically assigned a name at the time of its creation. This name consists of the object type name and an appended counter. The counter is displayed in brackets. Uniqueness of names within the namespace is taken into consideration here. This applies to the following objects:

Objects	Namespace
Hierarchy folder (THO)	Project
OS area ID and picture name for OS	Project
Charts (top charts and nested charts)	Program

Examples:

Folder name:	S7 program(1)
Chart name:	CFC(1)
OS picture name:	Picture(1)

Copying

The following applies when copying objects within or to other folders:

If the name is identical, the "Overwrite object" dialog box appears, with the options "Yes", "No" or "Rename":

- If you select "Yes", the object is overwritten and the name is retained. This only makes sense when copying data to another folder, since this would otherwise generate a zero operation.
- If you select "No", the object is copied and an additional counter is appended without delimiter.
  When the (original) object is copied once again, the appended counter is incremented.
  When you copy the copied version, an additional counter is appended.

Examples:

Chart:	CFC(1)	First copy	→	CFC(1)(1)
Chart:	CFC(1)	Repeated copy	→	CFC(1)(2)
Chart:	CFC(1)(2)	Copy	→	CFC(1)(2)(1)

- If you select "Rename", another dialog box appears. Here you can enter a new name or accept the default name automatically assigned in the input field.

Moving

The conventions for copying objects also apply when you move objects.

The response of objects described below differs to that of those described above:

Moving block instances

When you insert it in the chart, the block is automatically assigned a number as its name. This naming is sequential. Each new block inserted as a copy or via drag-and-drop from the catalog is assigned the next available higher number.

The following applies when you change the default name of this copied/moved block: Identical names are appended a number (without brackets). If the name being modified already ends in a number, this number is simply incremented.

Examples:

Block:	2	Copy	→	3
Block:	REG	Copy	→	REG1
Block:	REG1	Copy	→	REG2

Moving runtime groups

The response of copied runtime groups is identical to that of block instances when they are inserted in the run sequence. A sequential number is appended to the name without brackets.

6 What you should know about CFC

6.1 PCS 7 license information

Process objects (only relevant for CFC in PCS 7)

Process objects are all SFCs and all block instances that contain blocks which support reporting, operator control and monitoring. These are the objects that are transferred to the OS and require licenses. Driver block are not used as PO.

In the SIMATIC Manager , you can start a function that identifies all of the POs configured and booked in the Automation License Manager (ALM). The result is displayed in the "PCS 7 license information" dialog box. This allows you to check whether the existing "SIMATIC PCS 7 AS RT PO" license or the license you intend to order is adequate for your project, and the number of POs you can still add to your project.

Different process objects can use a varying number of licenses. How many licenses are needed depends on the functionality of the block type. You can view this information in the object properties of the block type in the module folder.

These objects are only entered in the count if they can be downloaded to an AS. Block instances in S7 programs without hardware assignment (at the project level or in libraries) are not considered.

Display PCS 7 license information

1. Selects the multiproject or the project.
2. Select the shortcut menu command "PCS 7 license information...".
   The "PCS 7 license information" dialog box opens. The right window shows the total of all process objects in the multiproject and the number of process objects in each individual project and station.
3. In the left window of the dialog box, select the "Process objects (AS RT PO)" license.

If you select an individual project, only the number of process objects in this project and the stations is shown.

You can find information on the scenarios for counting POs under: Count and booking of the PO licenses (Page 30)

6.2 Count and booking of the PO licenses

Basic sequence

The loader determines the process objects (PO) and the serial number of the CPU and Memory Card during the download. Aided by this number, a program is assigned to the CPU. A PO info is created in the ES data management for each CPU that is downloaded and records identifiers and the number of used PO licenses.

During the downloading process it is determined if and how many PO licenses for the current CPU have already been used. The number of POs that were identified during the last download is read from the PO info stored in the ES data management. The difference between the currently determined number of POs is compared with the number of licenses available in Automation License Manager (ALM). If the required POs are still covered by the license the difference is booked in the ALM and the download is executed.

If the current program contains fewer POs than the one previously downloaded, the number of available PO licenses is increased again through the download. If not enough licenses are available, the license violation generates a corresponding message that must be confirmed Now you may either terminate the download or continue just the same. The number of the required but unavailable licenses are recorded as a shortage. If you have purchased additional licenses, these missing licenses are included and registered in the ALM at the next download.

Licensing of the PO for the CPU 410-5H PA

The following additional properties apply to the licensing of POs for the CPU 410-5H PA:

- The number of POs is licensed by the hardware for the CPU 410-5H PA. It is not possible to download a greater number of POs than is licensed by the hardware into the AS.
- For the CPU 410-5H PA, the number of process objects (POs) that are booked in the Automation License Manager under the "AS RT PO" license are compared to the number of POs that are loaded in the CPU.
  If the number of booked licenses for POs in the Automation License Manager is lower than the number of POs loaded in the CPU 410-5H PA, there is insufficient licensing. This indication only applies if the CPU was loaded from this project and the POs were therefore booked in this project.
  The insufficient licensing is indicated by:
  – An entry in the diagnostics buffer of the CPU
  – Cyclic triggering of a corresponding message in SIMATIC WinCC.
    The message is always triggered immediately during the loading operation for which insufficient licensing is detected and when the difference changes due to the loading operation. After this, the message appears approximately every 6 hours.
    When the process objects are sufficiently licensed once again, for example due to the purchase of additional licenses, this message no longer appears. No "OUTGOING" message is generated.

Scenarios during the PO count

The following scenarios are considered:

- **First full download of a program**
  The loader determines that this program has never been downloaded. A PO info is created in the ES data management and assigned the CPU identification data along with the current number of POs. The current number of POs is booked in the ALM. This results in reducing the counter reading of licenses.
- **Downloading changes to a program on the same CPU**
  After comparing the data of the PO info the loader determines that this program has already been downloaded to the same CPU. The current number of POs is determined and the difference made up between value of the last download procedure. The difference is booked in the ALM. This results in reducing or increasing the counter reading of licenses. The stored number of POs in the PO info is replaced with the current number.
- **Moving a program to a different CPU**
  After comparing the data of the PO info the loader determines that this program has already been downloaded to a different CPU. A message is output which indicates that the number of POs stored during the last download can no longer be booked back if this download procedure continues. Here you can terminate the download process and book back the used PO licenses. Use the **Charts > Booking Back Process Objects** for selected chart folder.
  Requirement: The existing CPU must be connected.
  The PO info is updated with the data from the CPU that is currently connected when you continue the download. The current number of POs is determined and entered likewise in the PO info. This number is booked in the ALM. This results in reducing the counter reading of licenses.

- **Retiring a CPU**
  If you no longer wish to use a CPU then you can retrieve its used POs. Use the **Charts > Booking Back Process Objects** for selected chart folder.
  A comparison with the PO info is made to verify whether the same CPU that was used during the last download is connected.
  If this is not the case then a corresponding message appears informing that the PO licenses that were used during the last download can not be booked back by deleting the program in the connected CPU. The process must be terminated here in order to repeat it after the "correct" CPU is connected.
  A security check is made to verify the pending deletion of the program after the correct CPU is connected. The program is deleted after confirmation is made. The number of stored PO licenses is booked back in the ALM. Afterwards, the PO info is initialized; in other words, during the next download procedure, the program appears as if it were newly created.
  The PO info remains unchanged if you were unable to book back the POs because, for example, the license was no longer available. You can book back the POs at a later time after the license is available again.
  This function is terminated after a corresponding message appears if no CPU is connected or the one that is connected is not accessible.
- **Multiple use of a program in several CPUs**
  If the same program is downloaded to several CPUs then the number on the PO counter for each downloaded CPU is reduced by the current number of POs.
  After comparing the data of the PO info the loader determines that this program has already been downloaded to a different CPU. A message is output which indicates that the number of POs stored during the last download can no longer be booked back if this download procedure continues. Here you can terminate the download process. Otherwise the PO info is updated with the data from the CPU that is currently connected. The current number of POs is determined and entered likewise in the PO info. This number is booked in the ALM. This results in reducing the counter reading of licenses.

Notes on PO use

- **Download to S7-PLCSIM and test CPU:**
  No PO count is carried out here.
- **Deleting projects:**
  The CFC is not notified when you delete a project. Therefore you cannot book back any POs either.
- **Deleting chart folders or superimposed objects:**
  If you delete the chart folder, the S7 program, the CPU or the SIMATIC station, then this order to delete is passed on to the CFC. In this case a warning message appears that allows you to terminate the deletion process.

- **Defective CPU:**
  A new CPU that replaces a defective CPU is recognized as the previous download destination if the data stored in the PO info corresponds with the serial number of the CPU or the memory card. It is assumed that the "correct" CPU is connected for older CPU versions that do not allow a serial number query.
- **CPU replacement in a high availability system:**
  In a high availability system, if the memory cards used do not permit a query of the serial number and both CPUs are to be replaced, you should follow the procedure below.
  To avoid an incorrect booking of the PO of this high availability system, the CPUs may only be replaced one after the other. After the replacement of one CPU, the associated AS must be loaded again. Only then may the second CPU be replaced and loaded.

You can find additional information on process objects (PO) unter: PCS 7 license information (Page 29)

6.3 Drive for page file

Swap file on USB drive

A page file containing the most available storage space is automatically set up on the drive by the CFC. If the drive is a USB drive, you must take the following into account:

In order to prevent the storage medium performance from decreasing the performance of the computer during the swap activity, proceed as follows:

1. Open the System Properties dialog box by selecting **System > Hardware > Device Manager > USB Drive > Properties > Policies** in the Windows Control Panel.
2. Under “Write caching and safe removal”, select the “Optimize for performance” check box for the drive.
3. Before removing the drive, click "Safely Remove Hardware" in the info area of the taskbar.

6.4 Reaction of S7 CPUs to error

An S7 CPU will always continue to run in the event of number overflow/underflow or division by 0.

Reaction to such states can be determined via the block algorithm, by means of evaluation of the status bit in the status word.

You can find more information in the LAD/STL/FBD online help, for example, help on STL, statement list, floating-point functions, basic operations.

The response of library blocks is described in the corresponding "Troubleshooting" help topic.

6.5 Migration to the new CFC version

6.5.1 Migration to other CFC versions

Further processing

You can still process data created with CFC V3.x to CFC V7.x as follows in CFC V8.x:

- You can read and display the CFC data from previous versions with CFC V8.x.
- If you edit those data, the contents of each chart folder are converted at the first write access following a prompt for confirmation. This ensures compatibility.
- When upgrading ES data from versions 5.x or 6.x to V8.x, note the following if attributes of I/Os have been changed manually:

Note

Templates introduced in version 7.0 are also integrated in the project during the upgrade .

During this process, the attributes of I/Os are set to the values of the template, for example, "S7_m_c" = "TRUE" is set to "FALSE" when upgrading to the value of the template.

Therefore, if attributes of the I/Os have been changed manually in the original version, you need to check these attributes after the upgrade and adjust them if necessary.

General notes on compatibility

- A backwards conversion of V8.x data to previous versions is not possible.
- You cannot use previous CFC versions to edit data created with a newer CFC version.

6.5.2 Converting CFC library blocks to BOPs

Introduction

As of version V5.0, you can convert blocks previously stored in the CFC library to basic operations (BOPs).
The new BOPs are shown in the list "Blocks as BOPs" below.

You can achieve the following goals by using the basic operations:

- You improve CPU runtime.
- You require fewer DB numbers.

Projects created with CFC versions < V5.0 (old projects) are **not** modified automatically when you convert them to V7.x, since this would effect changes in the S7 program on the CPU.

Note

The "CFC Library" no longer contains the blocks now available as basic operations.

Migration

If you want to migrate old projects without modifying your existing program, proceed as follows:

1. Create a new S7 project program, including a chart folder and a CFC chart.
2. Copy the symbol table to the new S7 program.
3. Delete all blocks that are now shown as BOPs in the symbol table (you can find information on this in the "Blocks as BOPs" list below).
4. Double-click the chart icon.
   The CFC chart opens.
5. Use the menu command **Options > Customize > Compile/Download…** to open the "Settings for Compilation/Download" dialog box.
6. Set the number ranges according to the previous ranges and
   click "OK".
   The dialog box closes.
7. Use the menu command **Options > Customize > Copy/Move…** to open the "Settings for Copying/Moving" dialog box.
8. In "Connections to Addresses", select the option "Include" and
   click "OK".
   The dialog box closes.
9. Close and delete the newly created CFC chart.
10. Select all charts – not the chart folder! - of the old project and
    copy them to the new chart folder.
11. If your project contains user-specific blocks whose DBs are in the area reserved for other applications, copy them to the block folder manually.
12. Assign the oldS7 program name to the newS7 program.
13. Cut the new S7 program and
    paste it over the CPU of the old program.
    The new S7 program replaces the old S7 program.
14. Select the chart folder and then
    select the menu command **Edit > Compile** to recompile the program.

Result

The project has been converted to the new BOPs.

Exceptions

If you follow these steps the project will remain unchanged. Exceptions:

- Interconnections to runtime groups will be lost and must be redefined.
- The DB number assignment changes. When necessary, the AS-OS connection data must be transferred again ("Compile OS").

Blocks as BOPs

The CFC contains the blocks below as BOPs:

ABS_DI	EPS_DI	MIN2_DI	MUX4_I	ROR_W
ABS_I	EPS_I	MIN2_I	MUX8_BO	RS_FF
DI_DW	I_DW	MOD_DI	MUX8_DI	SHL_DW
DI_I	I_W	MOD_I	MUX8_I	SHL_W
DIV_DI	JK_FF	MUX2_BO	NEG_DI	SHR_DW
DIV_I	LIM_DI	MUX2_DI	NEG_I	SHR_W
DW_DI	LIM_I	MUX2_I	ROL_DW	SR_FF
DW_R	MAX2_DI	MUX4_BO	ROL_W	W_BY
DW_W	MAX2_I	MUX4_DI	ROR_DW	W_I

6.5.3 Migration of old projects to the enhanced runtime model

Introduction

Projects created with older CFC versions (< V6.0) can be easily migrated to the enhanced runtime model, which is available with CFC version V6.0 and higher, in the course of a migration.

Note

Migration is initiated after the user’s first write access and prompt confirmation.

Procedure

1. Runtime editor: Create a runtime group in a cyclic task for each chart in your project.
2. Assign the runtime group the name of the corresponding chart.
3. In the run sequence, move the blocks of each chart to the corresponding runtime group.

4. CFC Editor: Set the local installation pointer for each chart (block installation pointer). Do so by selecting a block at which you want to append all further blocks and select the shortcut menu command **Predecessor for Insertion Position**.
5. Runtime editor: Set the global installation pointer for the program (chart installation pointer). Do so by selecting a runtime group after which the runtime group of the next chart is to be inserted.
   Background: The "old" installation pointer previously used to refer to the “predecessor for insertion position" was transferred to the chart installation pointer during migration. It may, therefore, point to a task or runtime group that is not intended for the insertion of additional runtime groups (charts).

6.5.4 Migration of control blocks to the external view

Any SFC control block (SFC_CTRL) which exists in the project is deleted and replaced by the external view of the corresponding SFC chart when you migrate ES data from V5.x > V6.x or V7.x. The parameters and block interconnections of SFC_CTRL are activated in the external view.

Runtime response is not altered. The control block will be deleted from the run sequence and the SFC chart takes over its function.

Multiuser engineering 7

7.1 Multiuser engineering

Introduction

Multiuser engineering of projects or multiprojects is basically possible. This setup allows you to configure, test and commission target systems from different remote locations or from a PC network (multiuser engineering).

Basic options

- PCs are networked
  The project is located on the server or on a released drive of the PC and several users are able to simultaneously configure specific parts of the project.
  Detailed information pertaining to multi-user engineering and to consequences is available at: "Configuring in the network (Page 43)"
- PCs not connected to a network
  The project is split into several segments to be edited separately. and is subsequently merged to form a master project, for example, via a diskette or a ZIP drive.
- Projects in a multiproject
  The projects of a multiproject are stored on a central computer and can be split for distributed editing. The projects required for editing specific stations are transferred to the workstation of the respective author. Several projects can therefore be present on one workstation at the same time.
  After the project has been edited and returned to the multiproject, the system must synchronize all interproject data by means of the menu command **File > Multiproject > Align Projects** and perform all interproject functions required, for example, compile OS. Always execute this function when you prepare the multiproject for commissioning.
- Branching and merging project data
  As of CFC V6.0 you can branch the charts of an S7 program to form multiple working projects. The new concept of the V6.0 runtime model supports the branching and merging of individual project charts as well as the structuring of the run sequence by chart. You can find more information in:
  - Concept and use of installation pointers (Page 179)
  - Textual interconnections in branched and merged project data (Page 153)

Procedure for branching and merging project data

1. Copy a plant-relevant part of the project (single chart, several charts) to another project.
   Result: Textual interconnections to sources not included in the scope of the copy have been generated in this copy.
2. Edit the copied section separately (add, delete, change blocks and charts).

3. Copy the edited plant-relevant part back to the original project.
   Result: The system first deletes the charts with the same names from the original project. There are now textual interconnections in all charts that expect data from the deleted charts. The system then pastes the copies of the chart(s) into the project.
4. Close all "open" interconnections via menu command **Options > Make Textual Interconnections**.
   Result: The interconnections are made again in charts that were edited in another project and copied back to the source project, as well as in the charts for which textual interconnections were generated due to the deletion.

7.2 Configuring in the network

Configuring in the network

The system supports simultaneous multi-user access on your Engineering Stations to defined elements of projects which are stored on a central server or on the network drive of a station. Multi-user access is always possible outside the CFC.

NOTICE
Security information on configuration in the network When performing configuration in the network, make sure that the central server or a shared drive is accessible only to authorized persons. This should be ensured not only by measures on the level of the operating system and PC network. You can find information about access protection in PCS 7 in the section "Protecting projects/libraries with access protection" of the *Process Control System PCS 7; Engineering System* documentation.

Note

You can find more information about the engineering station on a central server in the section "Configuration in the network" of the *Process Control System PCS 7; Engineering System* documentation.

Feasible scenarios in multi-user mode:

Editing various charts from different chart folders

Different charts from different chart folders can be edited separately by several authors on different engineering workstations. Mutual interference should not be a problem.

Editing various charts from the same chart folder

Different charts from the same chart folder can be edited separately by several authors on different Engineering Stations There is less chance of mutual interference compared to working on the same chart. Conflicts cannot be entirely excluded because all charts access shared resources, such as the symbol table, run sequence etc.

Working on the same chart

If a chart has already been opened by an author and another author wants to open the same chart, this author receives a message. The author can then decide if he wants to open this chart or wait until the chart is closed again.

If several authors work on the same chart, they may interfere with each other's work. **This work procedure is therefore not recommended**.

The notification is enabled by default, but it can also be disabled. To do this, select the project in SIMATIC Manager and then select the menu command **Options > Charts > Notice when opening charts...**.

Reaction to a variety of actions

If access conflicts develop, the system always executes the action assigned the highest priority class. The action with the lower priority is aborted in this case. Priority of the actions:

Action	Priority
Short read actions	3
Short write actions	2
Prolonged read action	1
Prolonged write action	1

The first action started is executed if users initiate several actions which are assigned the same priority class.

- Short read actions (without resource usage) are:
  - Open chart
  - Open run sequence
  - Open dialog box

  When other short read actions are performed at the same time in addition, no adverse effect should be expected.
  Short or long write actions executed in parallel may lead to access conflicts, that is, the short read action is canceled.
- Short write actions (without resource usage) are:
  - Instancing, parameter assignment, interconnection etc.
  - Closing dialog boxes with OK

  When a short or long write action is performed at the same time in addition, access conflict may occur for the action latest started.

- Long read actions (with resource usage) are:
  - AS-OS data transfer (compile OS)
  - Displaying chart reference data

  When an access conflict does not occur immediately when a long read action is triggered, perhaps because a write action is already being performed, this action is performed without access conflict. Exception: see the notes relating to system behavior.
- Long write actions (with resource usage) are:
  - Optimization of the run sequence
  - Compiling and downloading (AS and OS)
  - Type import and update / cleanup of block types
  - Updating shared declarations
  - Generating module drivers
  - Readback
  - Creating block icons
  - Creating diagnostic screens
  - Updating the PH in a multiproject
  - Import / Export functions of the IEA and of the process object view
  - Import / Export functions of the ES objects of Version Cross Manager

  When an access conflict does not occur immediately when a long write action is triggered, perhaps because a write action is already being performed, this action is performed without access conflict. Exception: see the notes relating to system behavior.

Notes on procedures

- It is advisable to update the data at specific cyclic intervals (F5 function) if several authors are working on the same project. These updates ensure that all users work on an identical and consistent project database.
- It is not advisable to have several authors running long read or write actions in parallel. The authors should preferably coordinate their action schedules. It is hardly feasible to tolerate a situation in which one author is editing an SFC type, while a second author is setting out to compile the program, for example. The authors are responsible for coordinating simultaneous actions and for assessing the feasibility of such actions. Only one author is allowed to carry out prolonged actions at any chart folder. It is therefore advisable only to start engineering which involves prolonged actions if the database is consistent.
- Only **one** worker has access the data of an OS at any one time. The WinCC Explorer prevents access to the OS by more than one worker.

Notes relating to system behavior

- Any action which could not be completed because other applications outside the CFC are already accessing the resources will be canceled. This access may be initiated by STEP 7 applications such as the SCL / Block or Symbol Editor. The content of messages may differ depending on the triggering application. However, their significance is always the same: The action must be restarted after the other active application was completed or canceled.
  Example: You initiated a compiling session which is by now active over a longer period of time and the compiler wants to access the symbol table. If this symbol table is already in use by a different author, the compiler cancels the operation and generates a message informing you that you have to close the other application before you restart the action.
- If one user is editing data in offline mode while other users have activated the test mode, a message is output to these other users the next time they activate test mode to request recompilation and download of the chart to the CPU. The author is now responsible for deciding whether or not to activate the test mode. This may make more or less sense depending on the offline changes; authors should come to agreement in this regard.
- After receiving a message that the data is being used by another application, when a value is logged in or out for monitoring in test mode, it is not stored in the session memory. The logon or logoff must be repeated the next time test mode starts.
- When values monitored in test mode are no longer updated due to offline changes (because a block has been deleted, for example), the "#" character appears with a red background at the corresponding I/Os instead of the value.
- Any compilation initiated by an author while another author is editing parameters in test mode will be rejected with a message stating that the data is currently in use by a different application.

8 Starting and operating the CFC editor

8.1 Starting and closing the CFC editor

Starting CFC from the SIMATIC Manager

Proceed as follows to start the CFC editor:

Double-click the icon of the CFC chart you want to open.
The CFC editor starts and opens the chart.

Starting CFC from the Windows Desktop

Double-click the icon for the CFC editor (if available)

or

Select the program entry from the Start menu: **SIMATIC > STEP 7 > CFC**.

Note

A separate file save function is not available, since the CFC editor saves all file changes immediately. Note that you cannot undo or cancel changes in the CFC editor by closing the editor without saving.

You should always back up all data by copying the entire program to a backup project, You can then revert to old versions at any time. This also allows you to archive your complete configuration.

Closing CFC

Proceed as follows to close the CFC editor:

In CFC, select the menu command **Chart > Exit**.
The CFC Editor closes.

8.2 Operator controls and structure of the CFC editor

8.2.1 Working window

The editor lets you open any number of windows within the limits of Windows. Each chart window displays a CFC chart.

You can open multiple windows for the same chart, for example, to create well-arranged interconnections across sheet boundaries. You can open a clear view of the same chart in two windows by selecting **Window > New Window** > **Window > Arrange > Horizontally** or **Vertically**. The menu command **Arrange** > **Cascade** is unsuitable for this particular application.

The window in the foreground is always the active working window. This means that the menu commands and toolbar buttons have an effect on this window. Exception: Functions for opening and closing windows, for example.

The title bar in online mode is displayed in a different color (default setting: cyan). You can customize the default settings for the background and text in the SIMATIC Manager with the menu command **Options > Customize > "View" tab**.

CFC allows you to simultaneously display the chart window and the following windows:

- "Dynamic display"
- "Trend display"
- "Chart Inputs/Outputs" (interface editor)
- "Run sequence" (runtime editor)

8.2.2 Catalog of blocks, charts, templates and libraries

Opening and closing the catalog

You can insert blocks and charts selected from the catalog into your CFC chart.

There are several ways of opening and closing the catalog:

- Click the button in the toolbar

  or
- Select the menu command **View > Catalog**.
  or
- Use the <Ctrl+K >key shortcut.

There are several ways of opening the catalog:

- Select the menu command **Insert > Block/Chart (from Catalog)**
  or
- In the shortcut menu, select the command **Catalog**
  or
- Use the <F2> key.

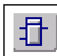
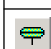
Positioning the catalog

You can drag-and-drop the catalog at its edge to any screen position. It is then shown as a separate window with a title bar.

If you move the catalog close enough to the right or left edge of the CFC window or double-click its title bar, it is docked accordingly and shares the workspace. When it is docked, you can also double-click the catalog's edge to open the catalog as a separate window and move it to any position.

Go to the catalog window

Within the catalog, you can navigate to the following catalogs:

	Block catalog
	Chart catalog
	Template catalog
	Library catalog
	If the project contains unplaced blocks, the catalog of unplaced blocks is also shown. This catalog is not available as by default.

Object search

Information on the search is available in the paragraph below "Options available in the lower part of the window".

Catalog of block types

You can select a block from the "All blocks" folder of a block family folder or from the S7 program folder and drag it to your chart. You can also select a block / new text / new chart and insert it in a free space in the chart using the <Return> key.

Blocks from the S7 program  are automatically imported to CFC when you insert them in the chart and also appear in a family of CFC blocks. The block icon indicates whether or not the block is already known to CFC, in other words, has been imported.

Blocks that were not imported look like this: 

Blocks in the S7 program are still unknown in CFC, even if a block of this type has already been imported. When you insert a block from the S7 program into the chart, the block is always checked first to determine if it has already been imported.

You can find imported blocks in one of the block families . A block that is not assigned to a family (indicated in the header) is found in the "Other blocks" folder. These blocks appear as .

In the "All blocks" folder, you can find all blocks of all block families (including "Other blocks"), sorted in alphabetical order. All imported blocks are therefore displayed twice, but in a different sorting order.

Note

Inserting blocks from the block families or from "All blocks" is the fastest way of inserting blocks in your chart, because no previous check is performed. You should therefore always use the imported blocks of the corresponding family.

You can also insert an empty text field into your chart using the "New Text" icon located above the block families, for example, when you want to enter a comment or notes on the configuration either via drag-and-drop or by selecting and applying with <Return>.

With the "New chart" icon displayed above the block families, you can insert a nested chart using drag-and-drop or by selecting and entering it with the <Return> key.

To open help on a block and the "New Text" or "New Chart" function, select the object and press the <F1> key to display the help topic.

Chart catalog

The chart layout in the catalog window can differ, depending on the settings made on the "Catalog" tab of the "Customize Layout" dialog box. You reach the area for these settings with the menu command **Options > Customize > Layout**.

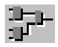
- When the "Select active chart" check box is selected, the chart symbol of the active chart is displayed as an open folder . The corresponding hierarchy branch is also opened if the active chart is nested. If the chart is not shown in the visible area of the catalog window after it is activated, the window is scrolled until the selected chart appears. When you close the hierarchy branch or move the visible area of the window, the selected chart is longer visible unless you activate it again, for example, by toggling the chart once.
- When the "Show with plant hierarchy" check box is selected in this dialog box, the charts are displayed with their plant hierarchy, in other words, with the hierarchy folders, provided the charts exist in a plant hierarchy.

  Note

  The catalog shows no entry if the "Show with plant hierarchy" check box is selected, although there is no plant hierarchy.

- If a plant hierarchy does not exist and the "Show catalog with plant hierarchy" check box is not selected, the charts are displayed in alphabetical order (as a tree).
- The + character in the box indicates charts that contain nested charts. Click this box to expand this tree level and displays the hierarchy of the chart.
  Note:
  The base chart is the chart displayed in the SIMATIC Manager . The nested charts contained within it are not displayed in the SIMATIC Manager .

You can select and drag-and-drop a CFC chart to your chart. The layout of the inserted chart is similar to that of a block. When this involves a chart with chart I/Os, the I/Os are also displayed. The representation of the chart also differs from that of a block by its chart icon:



When you insert the chart, it is copied to the chart folder along with any nested charts.

You can also hold down the shift key and drag-and-drop top charts to the chart. In this case, they are not copied but moved. The chart is now no longer at the previous location in the catalog. It is displayed in the hierarchy of the current chart.

You can also open charts directly in the catalog window. Select the required chart and then select the shortcut menu command **Open Chart**.

Template catalog

The base data library, lower-level PH folders and control module types are displayed hierarchically in the catalog.

You can move the control module types with drag-and-drop into the editor for technological connections.

Library catalog

Available block libraries are displayed in the catalog window by the icon . They represent a collection of additional block libraries, shown as block folders . The block types they contain are displayed as .

Click the plus box to open the libraries, select a block and drag-and-drop it into the chart. The block is automatically copied to the active S7 program and imported into CFC. It then also appears in the block catalog in a family of CFC blocks and in "All blocks".

The blocks are shown twice in the catalog but with different sorting:

- According to family (for example, FMT_PID in CONTROL)
  and
- In alphabetical order in the "All blocks" folder

The blocks are displayed with:

- Block name (for example, CTRL_PID)
- Block number (for example, FB 61)
- Block comment (for example, PID Control)
- Icon comment (if available)

If you require help on a block, you can select it and then display the associated help topic with the <F1> key.

Unplaced blocks catalog

Note

This catalog is only available if unplaced blocks actually exist.

Here you can select unplaced blocks and reinsert them in your CFC chart.

Unplaced blocks are arranged in the CFC charts in which they were originally placed. You can select a block and drag-and-drop it to the chart.

Options available in the lower part of the window

- Find
  In the catalog of blocks or libraries, you can enter a text and start the search by clicking the following button:
  
  The search starts at the object selected in the active catalog window. The search is performed through all folders until the specified object is found or until the search cycle reaches the selected object again.
  In the catalog of charts, you can perform a search for a chart, for example, for a nested chart. If the search result is positive, the function expands the tree level and highlights the selected chart.
  You will find detailed information on this topic in the section "Searching for objects in the catalog (Page 445)".
- Closing a folder
  If you want to close all folders, click the following button:
  

8.2.3 Menu bar

The menu bar is located at the extreme upper edge of the application window. Its functions affect the active working window. You can only select menu commands that have a feasible function in the current situation. For example, you can only select the menu command **Edit > Delete** when at least one object which can be deleted is selected, for example, a block, nested chart, connection line, or I/O.

Disabled menu commands are grayed out.

The bar displays only menu commands relevant for the selected target system.

8.2.4 Toolbar

The toolbar is located below the menu bar. It contains a series of icons that start frequently required menu bar functions, including a drop-down list for the selection of sheets/overview. You can see what function the button is associated with by positioning the cursor over the icon, but not clicking. A small box appears on the screen, displaying the designation (tooltip) of the function. More detailed information on the function is displayed in the status bar.

Clicking the button triggers the function. Disabled buttons are grayed out.

8.2.5 Status bar

The status bar is located at the bottom edge of the CFC window. It displays important information and the editor status. Its content changes according to operator input and editor status.

You can hide or show the status bar via the "View" menu.

Left area

The left area of the status bar displays context-sensitive information, for example, descriptions of menu commands, prompts or error messages.

Center area

- "Yellow triangle with an exclamation mark !" icon
  Display of an important message that is explained in text form to the right of this icon.
- Note text "TCiR: Download required !":
  This message indicates that at least one "type update in RUN" was performed and now a download to the CPU is required.

Right area

The right area of the status bar contains three fields:

- The right (larger) field shows the current mode (edit mode or test mode) and additional information:
  - In edit mode:
    Task, chart name and the predecessor for the insertion of blocks into the run sequence
    If the text "(Start)" is appended to the name of the runtime group, the insertion position lies at the beginning of the group, in other words, above the existing blocks, for example, OB 35 Group1 (Start).
  - In test mode:
    Test: Status of the CPU, breakpoint status, cause of stop
    The CPU status is also indicated by the background color:
    Light green = RUN
    Red = STOP
- The middle, smaller field shows the following chart information:
  - In the overview:
    Identifier of the chart partition (A – Z) and "Overview"
  - In the sheet view:
    Identifier of the chart partition (A - Z) and current sheet number (Sheet 1 - 6), possibly supplemented by the overflow page number
- If the SIMATIC Logon Service is installed and a user is logged on, the left field displays the user name.

8.2.6 Shortcut menu

A right-click opens a shortcut menu that contains frequently used menu bar functions. You can only select functions useful for the current application. All others are grayed out. You can access various shortcut menus, depending on the position of the cursor or selected object.

Examples of locations where different menus are called:

- Free area
- Block/nested chart
- Text field (when not in editing mode)
- I/O
- The chart partition tab
- Sheet bar entry (small field)
- Dialog boxes
- Windows of the run sequence, chart inputs/outputs and catalog

8.3 Keyboard and mouse operation

8.3.1 Shortcut keys for menu commands

Introduction

You can select any menu command via a corresponding shortcut using the <Alt> key.

Shortcut keys

Press the following keys in the order shown:

- <Alt> key
- The underlined letter in the name of the required menu (for example, <Alt>+<C> for the "Chart" menu - if the "Chart" menu is included in the menu bar). The menu opens.
- The underlined letter in the name of the required menu command (for example, <N> for the menu command "New").
  Any submenus of this menu command will also be opened. Repeat the actions described above until you have selected the menu command required.

The menu command is executed after you have entered the last letter of the shortcut key.

Examples

Function	Shortcut key
Chart > New	<Alt>, <C>, <N>
Edit > Go To, Sheet	<Alt>, <E>, <G>, <S>

8.3.2 Shortcut keys in the CFC chart

Overview

You can navigate in the CFC chart using the following keys:

- **Chart window**

Shortcut key	Function
<Tab>	A **single block** is selected: Moves the cursor to the next block (sequence defined internally, the user cannot intervene). **Multiple blocks** are selected: The selection is canceled and the cursor moves to the next block in the internal sequence. In the overview, the cursor moves to the next page. In the sheet view, the cursor moves only within the sheet; first to normally positioned and then to overlapping blocks.
<Shift+Tab>	Same as <Tab>, but reversed sequence.
<Arrow up>	**An input/output is selected:** Moves the cursor to the next higher input/output (if it exists; otherwise, the block will be selected).
<Arrow down>	**A block is selected:** Selects the first input (or first output, if only outputs exist). Input/output selected: Moves the cursor to the next lower input/output (if it exists).
<Right arrow>	**An input is selected:** Moves the cursor to the opposing output (if it exists; otherwise, to the last output).
<Left arrow>	**An output is selected:** Moves the cursor to the opposing input (if it exists, otherwise to the last output).
<Ctrl+Arrow up>	In the sheet view: Move to the next page up (e.g., 2 -> 1).
<Ctrl+Arrow down>	In the sheet view: Move to the next page down (e.g., 2 -> 3).
<Ctrl+Right arrow>	In the sheet view: Move to the next page on the right (e.g., 2 -> 5).
<Ctrl+Left arrow>	In the sheet view: Move to the next page on the left (e.g., 5 -> 2).
<Ctrl+Page Up>	Move to the previous chart partition, if one exists (e.g., B -> A).
<Ctrl+Page Down>	Move to the next chart partition, if one exists (e.g., B -> C).
<Ctrl+K>	Opens and closes the catalog.
<Ctrl+F6/Ctrl+Tab>	Switches between open windows.
<Alt+Page Up>	Shifts the visible part of the chart to the left.
<Alt+Page Down>	Shifts the visible part of the chart to the right.
<F6>	Toggles keyboard from the active chart to the catalog.
<Shift+F10>	Opens a shortcut menu for the selected object. Opens the shortcut menu for the empty space if multiple objects or no object is selected.
<Esc>	Cancels all existing selections in the chart (also multiple selections and the flashing function)

- **Interconnections**

Shortcut key	Function
<Ctrl+X>	Input: Copies the interconnection to the clipboard and deletes it at this input when you paste it to another selected input with <Ctrl+V>.
<Ctrl+C>	Input: Copies the interconnection to the clipboard
<Ctrl+C>	Output: Copies the output to the clipboard as a potential interconnection source

Shortcut key	Function
<Ctrl+C>	Sheet bar entry: Copies the interconnection to the clipboard
<Ctrl+V>	Input: If you have previously used <Ctrl+C> on an input/sheet bar entry or <Ctrl+X> on an input, the interconnection will be pasted from the clipboard to the selected input.
<Ctrl+V>	Input: If you have previously used <Ctrl+C> on an output, the input will be interconnected with this output.

- **Catalog window:**

Shortcut key	Function
<Ctrl+Page Up>	Switches catalog views, from right to left.
<Ctrl+Page Down>	Switches catalog views, from left to right.
<Return>	• In the "Blocks" window: Inserts the selected type into the active chart (new text, new chart or block instance) • In the "Charts" window: Opens the selected chart. • In the "Libraries" window: Imports the selected block and places the block instance in the active chart • In the "Unplaced Blocks" window: Places the selected block in the active chart • In the search box: Triggers the search.
<F6>	Toggles keyboard from the catalog to the active chart.
<Tab>	Switches between operating elements, from top to bottom or left to right (catalog views, search box, buttons)
<Shift+Tab>	Same as <Tab>, but in reversed direction.
<Arrow up> <Arrow down>	Moves from the selected to the next object in up or down direction
<Arrow right> <Arrow left>	Expands the hierarchy of the selected object and selects the first object at this level or moves one level up in the expanded hierarchy
<Page Up> <Page Down>	Selects the object at the upper or lower edge of the window. When actuated again, it shifts the visible area accordingly (if the full content was not visible)

- **Runtime editor window:**

Shortcut key	Function
<Tab>	Moves between the hierarchy window (left pane) and the detail window (right pane).
<Arrow up> <Arrow down>	Moves from the selected to the next object in up or down direction

Shortcut key	Function
<Arrow right> <Arrow left>	Expands the hierarchy of the selected object and selects the first object at this level or moves one level up in the expanded hierarchy
<Page Up> <Page Down>	Selects the object at the upper or lower edge of the window. When actuated again, it shifts the visible area accordingly (if the full content was not visible)

8.3.3 Cursor movement in the text editor

Keys for moving the text cursor

You move the text cursor within the text using the following keys:

Shortcut key	Function
<Arrow up>	One line up or one character to the left in a text consisting of only one line
<Arrow down>	One line down or one character to the right in a text consisting of only one line
<Right arrow>	One character to the right
<Left arrow>	One character to the left
<Ctrl+Right arrow>	One word to the right
<Ctrl+Left arrow>	One word to the left
<Pos1>	To the beginning of the line
<End>	To the end of the line
<Page Up>	To the previous page
<Page Down>	To the next page
<Ctrl+Home>	To the beginning of the text
<Ctrl+End>	To the end of the text

Additional information

You can find additional information in the section: International/German keyboard (Page 63)

8.3.4 Cursor movement in the menu bar/shortcut menu

Keys for moving the mouse pointer

You move the cursor in the menu bars or the shortcut menu using the following keys:

Shortcut key	Function
<F10>	Move to the menu bar
<Shift+F10>	Open the shortcut menu
<Alt+X>	Select the menu containing the underlined character X
Underlined character in the menu command	Select a menu command
<Left arrow>	Moves one menu command to the left
<Right arrow>	Moves one menu command to the right
<Arrow up>	Moves one menu command up
<Arrow down>	Moves one menu command down
<Return>	Execute the selected menu command
<Esc>	Exit the menu or return to text

Additional information

You can find additional information in the section: International/German keyboard (Page 63)

8.3.5 Cursor movement in dialog boxes

Keys for moving the mouse pointer

You move the cursor in dialog boxes using the following keys:

Shortcut key	Function
<Tab>	Move to the next input field (left to right and top to bottom)
<Shift+Tab>	Move one input field back
<Alt+X>	Move to input field that contains the underlined character X
Arrow keys	Move the cursor in a selection list
<Alt+Arrow down>	Open a selection list
<Spacebar>	Select an object or cancel this selection
<Return>	Confirms your entries and exits the dialog box (corresponds to the "OK" button).
<Esc>	Exits the dialog box without saving changes (corresponds to the "Cancel" button).

Additional information

You can find additional information in the section: International/German keyboard (Page 63)

8.3.6 Selecting text via shortcut keys

Key commands

You select text using the following keys:

Shortcut key	Function
<Shift+Right arrow>	One character to the right
<Shift+Left arrow>	One character to the left
<Ctrl+Shift+Right arrow>	One word to the right
<Ctrl+Shift+Left arrow>	One word to the left
<Shift+Home>	Text to beginning of line
<Shift+End>	Text to end of line
<Shift+Arrow up>	One line up
<Shift+Arrow down>	One line down
<Shift+Page Up>	To the previous page
<Shift+Page Down>	To the next page
<Ctrl+Shift+Home>	To the beginning of a file
<Ctrl+Shift+End>	To the end of a file

Additional information

You can find additional information in the section: International/German keyboard (Page 63)

8.3.7 Accessing help via shortcut keys

You call Help using the following keys:

Shortcut key	Function
<F1>	Opens help. If there is a current context, for example, a selected menu command, the relevant help topic will be opened. Otherwise the table of contents will be displayed.
<Alt+F4>	Closes the Help window and returns you to the CFC editor

8.3.8 Shortcut keys for configuring

Introduction

The next section lists the shortcut keys experienced users can deploy instead of the menu commands in order to accelerate access to specific functions.

Moving between chart sheets in the sheet view

Press <Ctrl> and <Left arrow>, <Arrow up>, <Right arrow>, <Arrow down> to browse the sheets if you are in the sheet view. The sheet is selected according to the arrow direction. If there is no sheet in this direction of the arrow the shortcut key has no effect.

Moving between the chart view and sheet view

Proceed as follows to toggle the views:

- In the sheet view:
  To change to the overview, double-click a free point on the sheet, in other words, not on a block, block I/O, interconnection, or the sheet bar.
- In the overview:
  To change to the sheet view, double-click a free point on the sheet, in other words, not on a block, block I/O, interconnection or the sheet bar. Displays the position at the middle of the sheet you have selected by double-clicking.

Interconnecting multiple block inputs to one block output

Proceed as follows to create the interconnection:

- Select the block output.
- Press the <Shift> key and select the block input.
  The interconnection is established and the block output remains selected.
- Press the <Shift> key and select the next block input.
- Select the last block input to be interconnected without pressing the <Shift> key.

Selecting a group of blocks

Proceed as follows to select a group of blocks:

- Hold down the <Ctrl> key and click any block required. Repeat this action until you have selected all blocks required. You can cancel a selection by clicking a selected block again, while holding down the <Ctrl> key.
- To select multiple blocks arranged in the same area, position the cursor at a point outside of the blocks, hold down the left mouse button, draw a lasso around the blocks and then release the mouse button. When you release the button, all the blocks within the lasso are selected.

Editing the object properties of a group of blocks

To edit the object properties, proceed as follows:

- Select the blocks you want to edit either by using the <Ctrl> key and clicking or by using the lasso function.
- Select the menu command **Edit > Object Properties....**
  The dialog for editing the first block opens.
- Make your required entries in the dialog box.
- Click "OK".
  You exit this dialog box and the dialog box for editing the next block opens.
- You can click "Cancel" to exit the dialog box without saving changes. This closes the dialog box. The set object properties are not applied.

8.3.9 Mouse operation

Overview

The mouse offers the following functions:

	Single click	**Double-click**	**<Ctrl> + click**	**Click + drag-and-drop**
Block header (also applies to the header nested chart)	Select object; deselect all other objects	Open “Properties – Block” or “Properties – Chart” dialog box	Select block/ nested chart	Move block/ nested chart
Block body	Select block; deselect all other objects	-	Select block	Move block
Runtime properties box in the block header	Select block; deselect all other blocks	Open “Run Sequence” window	Select block	Move block
Input, output	Select input/output	Open “Properties – Input/ Output” dialog box (or, with STRUCT, “Select Structure Element” dialog box)	Select block/ nested chart	Move interconnections (rewiring)
Nested chart body	Select object; deselect all other objects	Open nested chart	Select nested chart	Move nested chart
Connector	Signal tracking	-	Track signal	Draw lasso
Connection line	Signal tracking	Switch to sheet view or overview	Track signal	Draw lasso
Sheet bar entry (small field)	Field selected	Jump from a sheet bar with signal tracking; with addresses, open the "Modify interconnection to address" dialog box	Track signal	Draw lasso
Sheet bar entry (large field)	Signal tracking	As for small field	Track signal	Draw lasso

	Single click	Double-click	<Ctrl> + click	Click + drag-and-drop
Free area on the sheet	Cancel selection	Switch to sheet view or overview	-	Draw lasso
Free area of the sheet bar	-	-	-	-

Note

Note that you can also right-click for the corresponding shortcut menu commands, for example, “Jump from Sheet Bar” or “Go to I/O Module”, for the elements described above. You can find additional information on this in the following section: Shortcut menu (Page 54)

8.3.10 International/German keyboard

International keyboard	German keyboard
<Pos1>	<Pos1>
<End>	<End>
<PAGE UP>	<Page up>
<PAGE DOWN>	<Page down>
<Ctrl>	<Ctrl>
<Enter>/<Return>	<Enter>
<Del>	<Del>
<Insert> key	<Ins>
<BACKSPACE>	<Back>
<Up> <Down> <Left> <Right>	<Up arrow> <Down arrow> <Left arrow> <Right arrow>

9 Layout of configuration data

9.1 Charts

The chart represents the basic working unit of the CFC editor. Each chart has a name that is unique within a CPU. Each chart consists of up to 26 chart partitions (Page 66) each with 6 sheets (Page 67).

You can create charts either in the SIMATIC Manager or directly in the CFC editor.

You can insert additional charts into a chart. This can result in a nested chart with a nesting depth of up to eight charts. You can insert existing charts from the chart catalog or new charts from the block catalog using drag-and-drop or with the menu command **Insert > New Chart**.

9.2 Chart partitions

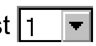
You can change the size of a CFC chart. You can increase the size of the chart by adding a maximum of 26 chart partitions (A to Z). You can decide whether to insert the new chart partition at the start or at the end of the current chart partition. Each chart partition consists of 6 sheets (Page 67).

At the bottom edge of the window you can select the chart partition you want to edit via the tab in the scroll bar. The status bar displays the letter of the currently active chart partition, including the sheet number, for example, "A/Sheet 2".

9.3 Sheets

Each chart (Page 65) (partition) consists of 6 sheets. In the CFC editor they are arranged in 2 columns, each containing 3 sheets. A sheet consists of a central work area and sheet bars with the sheet and chart references. In the work area, you can position and interconnect blocks.

Each sheet contains an internal grid pattern. Blocks can be moved in the sheet by distances determined by the grid pattern. This grid pattern determines the smallest possible distance between two connection lines.

The toolbar contains a drop-down list  from where you can select the sheet you want to edit. The status bar displays the current sheet number as follows: it includes the identifier (letters) of the chart partition, separated by a slash, for example, “A/Sheet 2”.

You can find additional information in the section: Display of interconnections in sheet bars (Page 79)

9.4 Overflow pages

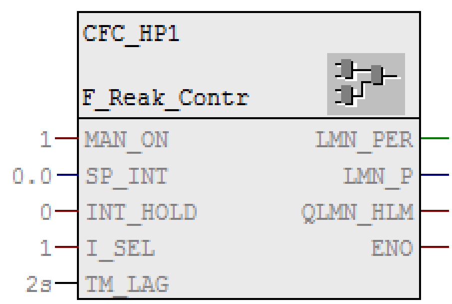
If so many interconnections between sheets are generated that there is insufficient space in the sheet bar for all the entries, an overflow page is created automatically. The overflow page represents an extension of the sheet bars and contains no additional objects.

9.5 Nested charts

A CFC chart can be inserted into another CFC chart (chart-in-chart technique). This allows the creation of hierarchy structures. Each nested chart can be opened and modified individually, the same as any other chart. The objects are placed on the work areas of the sheets.

A chart can be encapsulated for further use, in other words, it is assigned chart I/Os. You can individually determine which block I/Os are made available to the chart I/Os.

Nested charts are displayed in the chart as graphic objects, similar to blocks. The chart symbol is displayed in the chart header. Also included are the chart name and comments (if any comments exist).



You can also generate nested charts without chart I/Os. In this case, only the header and the empty body are displayed.

9.6 Blocks

Overview

Blocks are displayed in the chart as graphic objects. The objects are placed on the work areas of the sheets. Each block occupies a specific area. If blocks are positioned so that they partly or fully overlap other objects or the sheet bar, they are displayed in their normal size but without block I/Os and in another color.

The color for the display of overlapping blocks can be set in the "Color settings" dialog with the "Overlapping blocks" option.

Blocks can also be positioned across the horizontal line dividing two sheets. In this case, immediately below the dividing line, an additional block header is inserted that represents a copy of the first header. This allows blocks to be positioned that would not otherwise fit on one sheet.

The blocks are displayed in the chart with block I/Os that are identified at the block type as "visible" (S7_visible: = TRUE). In the object properties dialog of the block you can specify whether to show or hide the block I/Os.
Restriction: Due to space requirements, the chart can display no more than 160 inputs and 160 outputs of a block.

If the limit of 160 is exceeded, all other I/Os are hidden automatically. You can hide individual I/Os in the object properties of the block or the I/O.

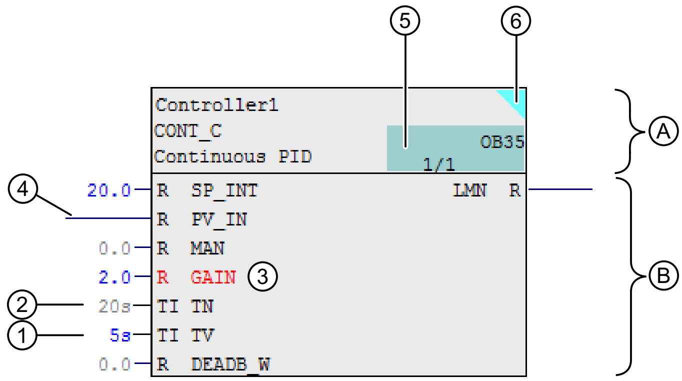
If you hide an interconnected I/O, the CFC responds as follows:

- For interconnections within a sheet, the interconnection of the connection partner is laid to the sheet bar. The text "INVISIBLE" is entered after the I/O name.
- For interconnections that go across sheets, the sheet bar entry is labeled with the text "INVISIBLE" following the I/O name.
- For textual interconnections and interconnections with shared addresses, to runtime groups and to the interface (chart I/Os), no sheet bar entry is generated. These interconnections can only be recognized in the object properties of the block, "Interconnection" column of the "I/O" tab.
- A colored triangle appears in the top right corner of the block header. (See item 6 in the figure below)

Block representation in the chart

The blocks are displayed in the chart graphically as "narrow" or "wide" blocks.

The figure below shows a block with a "wide" representation.



(A) Block header

(B) Block body

(1) I/Os with values that have been configured by the user are colored "blue" here, for example.
You can find more information in the description below.

(2) I/Os that have been assigned the default value are colored "light gray" here, for example.
You can find more information in the description below.

(3) The font color of the I/O name is changed (here "red", for example) for I/Os that have changed since they were last downloaded to the CPU.
You can find more information in the description below.

(4) Interconnected I/O

(5) Field for runtime properties

(6) Display that an interconnected I/O is set to "invisible".

The "wide" representation of a block shows the following:

- The block header (A):
  - Block name, block type and comment
  - Field for runtime properties of the block (5).
    You can find detailed information on this in the section "Display of the runtime properties box (Page 433)".
    Font color in the field for runtime properties of the block (5): Display of the download status of the block
    The font color in the field for runtime properties is used to indicate if a block has been downloaded to the CPU.
    The color can be configured in the "Color settings" dialog using the "Block insertion position/parameter (changed)" option.
    Example of the text color:
     - "Black" font means the block has been downloaded to the CPU.
     - "Light gray" font (depending on configuration) means, for example, that the block has not yet been downloaded to the CPU.

    Note

    Relevant changes to indicate the download status in the block header

    The font color in the field for runtime properties is used to indicate if a block has been downloaded to the CPU.

    The following changes have no effect on this status display:

    - Changes that have been made to the block, for example, changes to block I/Os (see 1, 2 and 3 in the figure above)
    - Changes to the corresponding chart or moving the chart to another runtime group are displayed on the chart or the chart folder.

  - Colored triangle (6) (optional):
    If there is an interconnected, hidden I/O, a colored triangle is displayed in the top right corner.
  - Block header is displayed in "Light green" color (default) if the block is a part of a technological type.
  - Block header is displayed in "Grayish light green" color (default) if the block is defined as a technological function.
- The block body (B):
  - "Inputs" and "outputs".
    The inputs and outputs are displayed as selectable fields that contain only the I/O name (narrow representation) or the I/O name and the data type (wide representation).
  - Free area
- The I/Os:
  - Outside the body, each interconnectable input or output has an I/O pin. Exception: Inputs that cannot be interconnected because they have been assigned the "S7_link := FALSE" attribute can be identified by the missing I/O pin.
    "Not interconnectable" in this context refers to an input you can neither interconnect to the I/O of a block or nested chart, nor to shared addresses. It is possible, however, to interconnect this I/O to its internal chart I/O, provided the CFC contains chart I/Os.

– Inside the body, block I/O is highlighted in "Light green" color if it is assigned to the technological interface of a technological type (control module, equipment module, and equipment phase).

– Inside the body, block I/O is highlighted in "Pink" color if the parameter value is set to be ignored for type update.

– Color representation of the status of the I/Os:
If a block or chart I/O has been changed, for example, due to configuration or interconnection, this status is indicated by a colored representation of this I/O (1)(2).
The font color of the I/O name is changed for I/Os that have changed since they were last downloaded to the CPU (3).
These color switches are reset after downloading the modified block to the CPU.
The colors can be set in the "Color settings" dialog. Open the dialog with the menu command **Options > Customize > Colors...** in the CFC Editor.

No. in figure	Status	Status indicated by:	Option of the "Color settings" dialog box
(1)	I/Os with values that have been configured by the user ("blue" here, for example).	Font color of the configured value/text in the block icon	"Parameter text (changed)"
(2)	I/Os that have been assigned the default value ("light gray" here, for example).	Font color of the configured value/text in the block icon	"Parameter text (default)"
(3)	I/Os that have changed since they were last downloaded to the CPU due to interconnection or configuration (here "red", for example).	Font color of the I/O name in the block icon	"Block insertion position/ parameter (changed)"

You can find additional information on this in the section "How to set I/O parameters (Page 134)".

You can set the width of blocks using the menu command **Options > Customize > Block/Sheet Bar Width...**. The settings apply to the chart folder that contains the active chart.

The settings are saved to the registry (CurrentUser) as a default, which you can use when you create a new chart folder. Each user can therefore define an individual logon setting.

"Narrow" representation of a block

The "narrow" representation of a block is similar to that of a block header of a "wide" block, but it includes the inputs and outputs.

The "narrow" block is displayed without fields or I/O names for the inputs and outputs and without a free area in the body. The I/Os are only represented by the I/O pin. The block width cannot be changed.

See also

Interconnections (Page 75)

9.7 Texts

Use the text field to enter a chart comment. You can position a text field in the chart just like blocks.

Drag-and-drop the text field "New Text" from the block catalog to the chart or insert it using the menu command **Insert > New Text** at a free location in the chart. You then open the selected text field by clicking on it. The text cursor is active and you can begin editing immediately. A line break is added automatically at the right edge of the box. If you enter more text than can be displayed in the box, the size of the box is not increased automatically; instead the text moves out of the visible area. You can make the entire text visible by increasing the size of the box manually.

To change the size of a box, click the box handles and drag with the mouse until the required size is reached. If you change the width of the box, the length of the text lines is automatically adapted.

When you open a text field, the text cursor is positioned where you clicked.

9.8 Interconnections

Overview

An interconnection is the connection between the following elements:

- A block/chart output to one or more inputs of another or the same block/chart or to the input of a chart I/O.
  The data types of the input and output must be compatible.
  Interconnected blocks or charts can be located in the same sheet, different sheets of the same chart or in different charts on the same CPU.
- A block output to a runtime group
- A block I/O to a chart I/O within one chart
- A block output to objects outside the CFC data management, for example, to shared addresses.

Another interconnection option is the textual interconnection. This represents a path reference to an interconnection partner that does not exist in the current chart folder.
You can find more information on this in section "Working with textual interconnections (Page 149)".

Color representation of a block or chart I/O after creating an interconnection

If a block or chart I/O has been changed, for example, due to configuration or interconnection, this status is indicated by a colored representation of this I/O.

You can find additional information on this in the section "Blocks (Page 70)".

Note

Each interconnectable input or output of a block is displayed with an I/O pin in the chart. Inputs that cannot be interconnected because they have been assigned the "S7_link := false" attribute can be identified by the missing I/O pin.
"Not interconnectable" in this context refers to an input you can neither interconnect to the I/O of a block or nested chart, nor to shared addresses. Textual interconnections are also impossible.
It is possible, however, to interconnect this I/O to its internal chart I/O, provided the CFC contains chart I/Os.

9.9 Layout of interconnections

Introduction

This section describes the following options when displaying interconnections:

- Block/chart interconnections
- Interconnections to shared addresses, runtime groups, chart I/Os, and textual interconnections
- Connectors
- Color display of interconnections
- Status display for dead time in signal processing of an interconnection

Block/chart interconnections

Layout of block/chart interconnections:

- When the blocks and/or charts to be interconnected are located in the same sheet, a connection line will be drawn. Reference entries will be generated automatically in the sheet bar if the objects are not located in the same sheet. Those references contain the identifier of the corresponding target I/O.
- Branches will be generated for a block/chart output that is interconnected to more than one input. A branch is indicated by a dot.
- On some target systems the interconnected binary inputs may have been inverted. This inversion is indicated by a dot.

Note

- Interconnections to a hidden I/O are laid to the sheet bar.
- A block with a hidden interconnection is indicated by a colored triangle in the top right corner of the block header.

Interconnections to shared addresses, runtime groups, chart I/Os, and textual interconnections

Sheet bar entries will be generated for interconnections to shared addresses, runtime groups or chart I/Os and for textual interconnections.

Note

If the interconnection begins at an I/O that has been subsequently hidden, there is no sheet bar entry. The interconnection can then only be made in the object properties of the block/chart, "Interconnection" column of the "I/O" tab.

Connectors

If no additional line can be drawn due to lack of sheet space, CFC will insert a connector at the block/chart I/O and enter a number in the sheet bar. The corresponding connectors will be assigned the same reference number. Multiple output interconnections that cannot be displayed will be assigned the same reference number. The differences in the layout of the connectors indicates the position of the I/O point:

24	Rectangle with point, white inner area	I/O point in the sheet
37	Rectangle filled with the color of the connection line	I/O point in the overflow page
37	Rectangle with point, filled with the color of the connection line	I/O point in the sheet and in the overflow page

If so many interconnections between sheets are generated that there is insufficient space in the sheet bar for all the entries, an overflow page is created automatically. The overflow page represents an extension of the sheet bars and contains no additional objects.

If an output contains multiple interconnections, some of the interconnections may only be displayed as a line. In such an event, a connector is attached to the line.

Using this connector technique, you can always display the full CFC structure, regardless of its complexity.

Color display of interconnections

The CFC editor displays interconnections, block/chart I/O points to inputs and outputs as well as the sheet bar entries in color.

- You can customize the colors of the interconnections depending on the data type using the menu command **Options > Customize > Colors...**. This feature enhances the chart layout and makes it easier to configure connections.
  - "Bool interconnection"
  - "Byte interconnection"
  - "(Double) word interconnection"
  - "(Double) integer interconnection"
  - "Real interconnection"
  - "Timer interconnection"

  The default color varies depending on the data type or group of data types.
- In the "Customize Display" dialog box, you can set whether the connection lines of an interconnection are shown in color or in black.
  You open this dialog box using the menu command **Options > Customize > Display**.

Display for feedback in the signal processing of an interconnection

If feedback occurs during the creation of an interconnection, this is detected and shown to the user.

- The following symbol is shown on the input side of the interconnected block at the respective interconnection in the CFC.

  The symbol is shown in the color of the interconnection line.

You can find additional information on feedback, the display and the possible user response in the section "Display of feedback in the signal processing (dead time) (Page 177)":

9.10 Display of interconnections in sheet bars

Layout

There is a sheet bar at the right and left margin of each sheet. The entries they contain are displayed in the sheet view.

The large sheet bar field has a variable width. You can set a width from 2 to 50 characters by selecting the **Options > Customize > Block/Sheet Bar Width** menu command in the dialog box. The settings apply to the entire chart folder.

A sheet bar entry is always generated when there is an interconnection to an object that is not located within the active sheet. The entry consists of two fields, each with two lines. The type of connection determines the field content.

Large field

The large field contains the reference to the linked object:

- Interconnection between blocks:
  Line 1: Chart name*), block name *)
  Line 2: I/O name and, if available, comment of the I/O (may be abbreviated)
  The chart partition and sheet number are given in parentheses after the chart name for sheet-wide interconnections.
  The technological path or, when the PH is not available, the component path is given for an interconnection to an I/O located in another CPU.
- Interconnection to shared addresses:
  Line 1: Symbol and/or absolute address value based on symbol table **)
  Line 2: Comment from symbol table **)
- Interconnection to a runtime group:
  Line 1: Name and, if available, comment for the runtime group
  Line 2: "ENABLE", task name
- Interconnection to unplaced block
  Line 1: Block name and text: (NO POSITION)
  2. I/O name and, if available, comment of the I/O
- Interconnection of I/Os (block/chart) to chart I/Os
  Line 1: I/O name and, if available, comment
  Line 2: I/O type and data type
- Interconnection to hidden I/O
  Line 1: Chart name*), block name *)
  Line 2: I/O name and text: (INVISIBLE) and, if available, comment of the input/output (may be abbreviated)

*)	If the chart is used in a plant hierarchy (PCS 7), the chart and block names are preceded by the hierarchy path name. The start of the path name is truncated if there is not enough space.
**)	The symbol table is not available in all target systems.

Note

Depending on the set sheet bar width, it may happen that the full text cannot be displayed, i.e., it will be truncated according to the default number of characters. However, the tooltip box will show you the full text when you position the cursor on the field.

Small field

The small field of the sheet bar entry is used as follows:

- The field contains the number of the connector reference when the connection line cannot be drawn to the sheet bar due to the lack of sheet space.
- A colored triangle indicates the connection type.

ID	Interconnected to ...
(empty)	Block I/O / chart I/O (external)
White triangle	Chart I/O (internal)
Blue triangle	Shared addresses
Red triangle	Runtime group
Yellow triangle	Textual interconnection
Green triangle	Connection to another CPU

You cannot change the width of the small field.

Entries in the overflow page

An overflow page will be generated if the sheet bar cannot accept additional reference entries for interconnections to other sheets.

Information on the interconnected I/O (origin) is also displayed next to the entries in the sheet bar of the overflow page (small and large field).

9.11 Tooltips

You can display a tooltip about certain parts of the CFC chart by positioning the cursor over the relevant items:

Position of the cursor	Information
Block header, block body	Block type, block comment
Chart header, chart body	Type (always "Chart"), chart comment
Runtime properties box in the block header	Full task name, name of runtime group, if available
Block I/O / chart I/O	I/O name I/O type Data type Value (only for I/O which is not interconnected) I/O comment
Sheet bar	Full sheet bar entry (may be truncated if PH path is too long)

The following applies to the test mode:

- The information is actively retrieved from the CPU by placing the cursor on an I/O or the connection line even if the I/O has not been registered for monitoring. The cursor here functions like a "test prod" of a measuring device. This requires, however, for the I/O to be updated online (no unconnected FC input, for example). Updating occurs in a one second cycle, regardless of the monitoring cycle that is set.
- The value is substituted with the online value and the background color of the tooltip changes from light yellow to yellow, as it does with monitored I/Os.
- The output value is displayed for an interconnection.
  Exception: With interconnections to shared addresses, the input value is displayed.
  Inversions are taken into account in this case.

9.12 Views

9.12.1 Views in CFC

Views

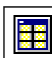
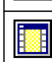
The CFC editor provides two different views for inserting and editing blocks/nested charts:

- The overview is useful for copying/moving objects and inserting large blocks.
- Because some details (I/O names, for example) cannot be displayed in this view, certain functions are only available in the sheet view.
  You can find additional information about this in the following sections:
  - Overview (Page 82)
  - Sheet view (Page 83)

Toggling between views

You have the following options for toggling between views:

- Select the menu command **View > Overview** or **View > Sheet View**.
  The CFC will switch to the selected view
  or
- Select the button for overview or sheet view in the toolbar.

	Overview
	Sheet view

or

- Double-click a free location in the chart
  or
- Select the sheet number in the combo box 1 to open the required sheet view or the asterisk (*) to change to the overview.

9.12.2 Overview

The "Overview" chart view

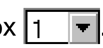
The overview (chart view) is useful for copying and moving blocks to or from nested charts and for inserting large blocks.

It is also possible to make interconnections of block I/Os, even beyond sheet margins. The information about I/Os (for example, name, I/O type and data type) that cannot be displayed in the chart view can be shown as a tooltip.

Changing to the "Chart view"

You have several options for changing to the "Chart view":

- Select the menu command **View > Chart View.**
  or
- In the toolbar, select the icon:
  

  or
- Double-click a free point in the chart in the sheet view
  or
- Select the asterisk (*) in the combo box .

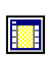
9.12.3 Sheet view

Sheet view

Depending on the resolution of your screen and the display size you have set, the sheet view displays the full sheet or only parts of it. Use the scroll bars to scroll the display.

Changing to the "Sheet view"

You have several options for changing to the "Sheet view":

- Select the menu command **View > Sheet View.**
  or
- In the toolbar, select the icon:
  

  or
- Select the required sheet number in the combo box of the toolbar
  or
- Double-click a free area of the chart in the overview
  or
- Select the sheet number in the combo box .

10 Creating runtime structures

10.1 Working with charts

10.1.1 How to create and delete charts

Requirement

In the SIMATIC Manager , you have created a project with an S7 program including a chart folder.

Creating a chart

You usually create a chart with the SIMATIC Manager . Proceed as follows:

1. Open a chart folder in the project.
2. Select the menu command **Insert > S7 Software > CFC**.
   The chart is inserted and assigned a default name by the system that you can change to suit your purposes.
   You can find additional information on this in the following section: Automatic naming (Page 27)

You can also create a chart directly in the CFC editor. Proceed as follows:

1. In the CFC editor, select the menu command **Chart > New…**.
2. In the dialog box, select the project and the chart folder.
3. Under "Object name:" enter the name of the new chart.
   The maximum length of the chart name is 22 characters (illegal characters: **\ / . " %**).
4. Click "OK".
   The dialog box closes.

Result

The chart has been created.

Creating a runtime group automatically

When you create a new chart, a new runtime group is automatically generated and inserted into the run sequence according to the chart installation pointer. The name of the runtime group corresponds to the chart name.

There is a certain amount of interdependence between the automatically generated runtime group and the chart until the runtime group is modified. Modifications are, for example, renaming and/or subsequent insertion of other blocks into this runtime group.

This interdependence ensures that the name of the runtime group is identical to the chart name and that it is automatically renamed when the chart is renamed. Furthermore, the runtime group is also deleted when the chart is deleted if it is empty as a result.

When this interdependence disappears due to modifications to the runtime group, the automatically generated runtime group responds in the same way as any other manually inserted runtime group.

Nested chart

You generate a nested chart when you insert a new chart into an existing chart. You can find additional information in the section: Creating nested charts (Page 95)

Deleting a chart

You can delete a chart similar to other objects in the SIMATIC Manager :

- Select the chart and press <Del>.

10.1.2 How to open charts

Procedure

You usually open a chart in the SIMATIC Manager:

1. First, select a project. Secondly, select the S7 program folder and open the chart folder.
2. Double-click the required chart.

The chart opens and the CFC editor started.

Additional options for opening a chart

The "Chart" menu displays the last four charts edited and closed in the form of menu commands. Select one of these menu commands to open the corresponding chart or to display it if it has been opened already.

You can open a chart that is not displayed in the "Chart" menu as follows:

1. Select the **Chart > Open...** menu command or click the button
   
2. In the dialog box, select the project and the S7 program folder.
3. Open the chart folder.
4. Double-click on the required chart
   or
   select the required chart and click on the "OK" button.
   The chart opens.

You can select a chart in the chart catalog and open it with the shortcut menu command **Open** or by pressing <Return>. The chart opens.

In addition to opening nested charts in the chart catalog, you can also open them as follows:

- Select the nested chart, then select the **Open** menu command, in either the "Edit" menu or the shortcut menu.
- A double-click on a free area in the body (not on the header or an I/O!) also opens the chart. By repeating this, you can work down to the lowest chart level in the tree.

From a nested chart you can open the parent chart of the current nested chart:

- Select the menu command **Open Parent Chart** from the "Chart" menu or shortcut menu. The parent chart opens at the position of the nested chart. The nested chart is highlighted.

10.1.3 Navigating in the chart

Introduction

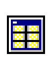
Several functions help you browse the chart. The following options are available:

- Menu command **Edit > Go To > ....**
- Double-clicking
- buttons and combo boxes on the toolbar

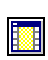
Toolbar Icons

You can change to the respective view using the icons.

- Overview

- Sheet view

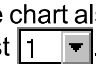
The last setting determines the size of the new view.

Double-click

Double-clicking on a free area changes the view. You change from the overview to the sheet view and vice versa.

When you change to the sheet view, the position of the cursor determines the center of alignment for the view.

Drop-down list for sheet and overview

The toolbar of the chart also allows you to change to the sheet required and to the overview via the drop-down list 

.

Chart partition tabs

The chart partitions are displayed on a maximum of 26 tabs located next to the scroll bar at the lower edge of the window. You can use these tabs to change to another existing chart partition by clicking.

Go To

The menu command **Edit > Go To >** opens a submenu which provides the following navigation options:

- **Insert Position**
  This function starts the runtime editor and displays the block selected in the chart in the run sequence.
  This menu command will be disabled if more than one block or no block is selected in the chart.
- **Next Insert Position**
  Use this function to find further insertion positions for the selected block in the run sequence.
  This menu command is only enabled after you have started the runtime editor and when a block/SFC chart has been selected in the run sequence.
- **Chart**
  Use this function to open the chart that contains the selected block. This block will be highlighted in the open chart.
  This menu command is only enabled after you have started the runtime editor and when a block/SFC chart has been selected in the run sequence.
- **Track Signal**
  You can use this function to track the signal of a selected chart I/O of a nested chart to the I/O which is internally interconnected to the I/O of this chart. Signal tracking means that the nested chart will be opened and the displayed interconnection and sheet bar entry will flash.
  This menu command is only enabled if you have selected a chart I/O that is internally interconnected with a block/chart I/O.
- **I/O Module**
  After you have selected the field of a shared address from the sheet bar, use this function to call "HW Config". An error message will be output to you if the entry does not refer to a module or if the address could not be found.
- **Block Type**
  Use this command to jump from the selected block instance to the corresponding block type.
  If the source file of the block is included in the project, the relevant programming tool (LAD/STL/FBD or SCL) is opened. You can use this programming tool to edit the block type.
  Otherwise, LAD/STL/FBD is opened, enabling you to read the block (for example, the system attributes of the I/Os).
- **Jump Back**
  With this function, you return to the sheet you exited by clicking the sheet bar.
  This menu command is only enabled if the window displaying the original chart is still open.
- **Next Overflow Page**
  This function lets you jump from the current overflow page or from the original sheet to the next overflow page.
  This menu command is only enabled if an overflow page exists.

- **Previous Overflow Page**
  With this function, you can jump from the current to the previous overflow page or from the first overflow page to the original sheet.
  This menu command is only enabled within an overflow page.
- **Original Sheet**
  Use this function to return from an overflow page of a sheet to the original sheet (sheet with blocks/nested charts).
  This menu command is only enabled if you are currently in an overflow sheet.
- **Chart Partition...**
  This function displays a dialog box with a list of all chart partitions. By selecting the required chart partition letter (A – Z) and clicking "OK" you can change to the required chart partition.
- **Sheet...**
  This function opens a dialog box that contains buttons that allow you to select one of the 6 sheets. Click the appropriate button to jump to the required sheet.

Jump from Sheet Bar

If an I/O is interconnected with another I/O via the sheet bar or is located on another sheet or chart, you can jump to the interconnection partner.

- Single interconnection
  Select the interconnection (I/O / sheet bar entry / interconnection line) and select the shortcut menu command **Jump from Sheet Bar**. You can also double-click the sheet bar entry.
  The corresponding sheet that contains the interconnection partner opens. The target chart is opened if the interconnection partner is located in another chart. The interconnection flashes (connection line and sheet bar entry).
- Multiple interconnection
  If this is an I/O that contains multiple output interconnections, a dialog box appears showing a list of the interconnected I/Os before the sheet bar jump is executed. There you can double-click the I/O required or you can confirm a selection with "OK" in order to jump to the sheet/chart of the interconnection partner.

Signal tracking

Proceed as follows to track signals:

Signal tracking from within a nested chart

1. Within the nested chart, position the cursor on the sheet bar entry.
2. Select the shortcut menu command **Jump from Sheet Bar**.
   The higher-level chart is opened and the interconnection flashes.

Signal tracking into a nested chart

You can find information on this in the above section “Go To” under “Track Signal”.

Signal tracking at the overview level

Click an interconnection.
All the interconnection lines associated with a chart partition flash on the following pages:

- Overview page
- All six sheets
- All overflow pages

10.1.4 Copying/moving charts

Introduction

Copying entire charts allows you to duplicate or move tested structures or substructures, even to other CPUs.

Copying/moving

Including resources in the copy

When you copy/move charts, the resources will also be copied, provided they do not already exist in the destination. The following elements are resources:

- Block types, FBs and FCs, including symbols
- System attributes
- Called blocks of multiple instance blocks

Textual interconnections

When you copy/move a chart to another chart folder, interconnections that exceed the chart boundaries are implemented as textual interconnections. These open textual interconnections can be closed again. They then form real block interconnections, provided you copy/move the chart and the interconnection partner to this chart folder, or once you have copied/moved the corresponding chart back to its original folder.

This includes textual interconnections of the chart you want to copy/move.

Peculiarities of nested charts: When you move or copy nested charts, the internal interconnections to the chart I/Os are not converted to textual interconnections, that is, they will be lost.

You can find additional information in the section: Working with textual interconnections (Page 149)

Shared addresses

When copying charts, remember that interconnections to shared addresses are also copied, depending on the default setting. You can change this setting in the "Settings for Copying/Moving" dialog box. You can open this dialog box with the menu command Options > Customize > Copy/Move....

Runtime groups

When copying/moving charts, interconnections to runtime groups are not copied. No textual interconnections are generated.

System attributes

When charts are copied or moved, the system attributes of the copied/moved blocks are compared to existing blocks in the destination. You may abort the copying/moving operation based on the results of the comparison. The causes of this are described in the following section: What you should know about copying blocks (Page 128) (under "Checking the system attributes").

Copying/moving charts to a different CPU/library/project

When you copy charts to other CPUs, remember the following points:

- If a block type in the destination CPU is incompatible with the block type you are copying (number, order, name, and data types of the block I/Os), the chart will not be copied.
  You are shown a log with a list of relevant block types.
  In this case, before you copy the chart again:
  - You must copy the relevant block type either to the block folder of the source program or to the block folder of the destination program.
  - Select the block type in the “Chart Folder” box using the menu command **Options > Block Types...** and click the “New Version” button.
    A central block type change is made.
- The copied blocks will maintain the runtime properties of the blocks from the source CPU. Therefore, the blocks are installed in the run sequence as they were installed in the source CPU.
  If no task with the same name exists on the destination CPU, a log displaying the missing tasks is output and the copy operation will be rejected.

Note

We do not advise the copying of complete chart folders, since the resources (e.g., FBs, FCs, shared DBs, symbols) are not included and program inconsistency might develop.

If you do not want to copy the complete program, you can select and copy all the charts in the chart folder. In this case, the resources are also copied.

Copying the entire program, the CPU or a station does not lead to any inconsistency.

If a chart is empty, the chart runtime group is not copied or moved.

Charts on different target systems

You can copy and move CFC charts between different target systems. Once again, the block types used on both target systems must be identical, since the block types are not included. If the block types are not identical, copying of the charts would be rejected.

10.2 Editing charts

10.2.1 Adapting chart properties

Use and call

In the "General" tab of the "Properties CFC" dialog box, you can set general properties such as the chart name, author and comment for the active chart. In the "Version" tab, you can change the version number, for example, after changing the chart.

In CFC, you can display this dialog box using the menu command **Chart > Properties...**.

Note: The dialog box opens automatically with the "Version" tab, if versioning has been activated in the project properties and the chart is closed after a change.

Content of the "General" tab

The "General" tab contains the following input and output fields:

- **Name**
  You can display and edit the name of the active CFC chart here. The name must be unique in the CPU. This is checked by the system.
  The maximum permitted name length is 22 characters. The following characters are prohibited: / \ . " %.

  Note

  When assigning names, please consider that the variable name cannot be longer than 128 characters for the OS compilation. This name is made up of the name of the folder in the hierarchy path, the chart name, the block name, the delimiter (period), and the I/O name.

- **Project path**
- **Technological path**
- **Storage location of project**
  You cannot change the path displayed in this dialog box. You can only do this from the SIMATIC Manager. The technological path is only available if a plant hierarchy has been configured in the project and a hierarchy folder has been assigned to the CFC chart.
- **Author**
  The author of the chart.
- **Date created**
  Date of creation
- **Last modified**
  Date of most recent modification If a chart has been modified, the date of most recent modification applies to all other charts in the same chart folder because there may be interconnections from one chart to another. Only one global "last modified" date is, therefore, applied to the chart folders.

- **Comment**
  Here, you can enter a comment that is included on your printed CFC chart.
- **"Write-protected" option**
  Use this option to enable or disable write protection for a chart. You cannot edit the write-protected CFC chart in the CFC editor. Because of the write protection, you can no longer alter any SFC instances in the chart using the SFC editor.

Content of the "Version" tab

In this tab, you can change the version number of the chart.

Version:

Major and minor versions can be set separately with the arrow keys. Version change is deactivated for write-protected charts.

You can set the versions from 0.0001 to 255.4095. A new chart always starts with version number 0.0001. No number less than the previously saved version can be set.

If versioning is activated in the properties of the project, the dialog box automatically opens with the "Version" tab when the chart is closed after a change.

Data version:

Displays the software version used to create or last modify the program. The data version depends on the product version. The data version is determined by the database and displays the current status of the data structure.

10.2.2 How to insert and delete chart partitions

Introduction

You can always expand the CFC chart by additional chart partitions, if its size is insufficient. The tab level with the bottom scroll bar can be used to select the chart partition required.

Insert

When you insert a chart partition, you can decide whether the new chart partition is inserted before the current chart partition or whether it should be appended as the last chart partition. You have two options:

Before the current chart partition:

- Select the menu command **Insert > Chart Partition > Before Active Partition**
  or
  open the shortcut menu of the chart partition tab and select the **Insert Before Active Partition** menu command.

As the last chart partition:

- Select the menu command **Insert > Chart Partition > After Last Partition**
  or
  open the shortcut menu for the chart partition tab and select the menu command **Insert Chart Partition After Last**.

A chart can contain up to 26 chart partitions. They are identified in alphabetical order (A-Z). The alphabetical identifier of the existing chart partitions changes if additional chart partitions are inserted before it.

If, for example, the "CFC1" chart consists of a single chart partition, it is assigned the ID letter "A". If you insert an additional chart partition before this chart partition, the new "primary" chart partition is assigned the "A" identifier, while the identifier of the previous partition changes to "B".

Depending on the view (sheet or overview), the ID of the current chart partition is displayed with the sheet number or "Overview" in the status bar.
Example:

B/Sheet 3 **B/Overview**

Delete

You can only delete the current chart partition.

- Select the menu command **Edit > Delete Chart Partition** or, in the shortcut menu for the chart partition tab, select the menu command **Delete Chart Partition**.
  The current chart partition is deleted if it is empty. If the chart partition contains any objects, you are prompted to confirm deletion.

10.3 Creating nested charts

10.3.1 Creating nested charts

Introduction

You can insert a CFC chart into another CFC chart. This allows you to develop a program whose modular segments can be standardized for reuse, with a structure according to your system requirements.

You can open and modify the (nested) charts inserted in the parent chart in edit mode and you can operate and monitor them in test mode on the CPU.

The chart containing the nested charts is the top chart. Only the top chart can be seen in the SIMATIC Manager . The maximum nesting depth of nested charts is 8 (top chart + 7 levels of nested charts).

Chart-in-chart

You can drag-and-drop a new nested chart from the CFC catalog of blocks (symbol: New chart) to the current chart for further editing. Execute the menu command **Insert > New Chart** in order to find a free location in the chart (in the sheet view only in the current sheet) and insert the new chart at this position. The search for this free area is performed horizontally, line by line, from left to right, and from top to bottom (in the overview of all sheets).

In order to handle charts in the same way as CFC blocks, the chart catalog displays the charts in a tree view. From this location you can drag-and-drop a chart to the currently open chart. When you insert the chart, it is copied to the chart, along with all nested charts, if it contains any.

Note

Remember that the chart you have inserted into the current CFC chart via drag-and-drop from the chart catalog is a copy. This nested chart no longer has any relationship to the original chart. Modifications that you make in one of the charts only affect that chart.

Moving a chart

You can also hold down the shift key and drag-and-drop top charts to the chart. In this case, they are not copied but moved. In the catalog, the chart no longer appears at its previous location. It is shown in the hierarchy of the active chart instead.

Interconnecting

You can interconnect nested charts with other nested charts in the chart, if they have chart I/Os, with blocks or shared addresses, or establish textual interconnections.

Navigating in a chart

Opening nested charts:

1. Select the nested chart.
2. Open the shortcut menu and select the command **Open**
   or
   select the menu command **Edit > Open**
   or
   double click on an open space in the body (not on the header or an I/O!).
   The chart opens.
   By repeating this, you can work down to the lowest chart level in the tree.

Moving up through the hierarchy as far as the top chart:

1. Activate the nested chart.
2. Open the shortcut menu and select the command **Open Parent Chart**
   or
   select the menu command **Chart > Open Parent Chart**
   or
   double click on an entry in the sheet bar (not on shared addresses).
   The parent chart opens.

Opening a chart in the chart catalog:

1. Select a chart in the catalog.
2. Select the shortcut menu command **Open**
   or
   press the <Return> button.

The active chart is identified in the catalog by its black border.

Copying a nested chart to another CPU

You can copy a nested chart and paste it into the chart of another CPU. The installation position of the "Predecessor for Insertion Position" of the target chart applies to the blocks it contains. Runtime groups are not copied.

You can find more information about nested charts in the following sections:

Copying/deleting nested charts (Page 96)

How to replace nested charts (Page 97)

How to create a chart with chart I/Os (Page 98)

10.3.2 Copying/deleting nested charts

Copying/moving within a chart or to other charts

You can copy or move nested charts in the same way as blocks.

Copying to the chart folder

You can first copy a nested chart into the chart folder if you want to delete it from its parent chart, but not from the chart folder (you may have modified it and want to use it later at a different point). Proceed as follows:

1. Open the nested chart to be copied by selecting **Edit > Open**
   or
   by selecting the **Open** command from the shortcut menu of the nested chart.
2. Once the chart is open (active), select the menu command **Chart > Copy to Chart Folder**.

The chart will be copied to the chart folder as a top chart. If a top chart already exists under the same name in this chart folder, a new name will be automatically assigned to it (with a message), provided you have not modified it before you copied it.

Deleting nested chart

You can delete the nested chart from its parent chart as follows:

- Select the chart and press <Del>.
  or
- select the shortcut menu command **Delete**
  or
- select the menu command **Edit > Delete**.

10.3.3 How to replace nested charts

Introduction

You can replace a nested chart with another. In this case, the interconnections of the “old” chart are maintained, if possible.

Application

The “Replace” function can be very useful in situations where you have configured the nested charts as encapsulated functions for different applications. Such an application might be:

You have encapsulated a (partial) function in a nested chart and interconnected it in this chart. For example, this partial function can be a control for a ventilation unit of which several variants are available for different fields of application. These variants can be replaced within the global structure, without you having to modify the interconnections.

Procedure

The chart that is to replace a specific nested chart is located either in a library, the current chart folder (as a top chart) or in another chart of the chart folder.

1. Open the chart that contains the nested chart you want to replace.
2. Open the chart or library catalog; expand the hierarchy tree, if required.

3. Drag-and-drop the required chart from the catalog onto the nested chart you want to replace.
4. Release the mouse button when the cursor is positioned exactly over the chart you want to replace.
   You are prompted to confirm the action.
5. Click "OK".
   The nested chart will be replaced, provided the cursor was positioned within the chart you want to replace.
   Click "Cancel" to abort this operation.

10.3.4 How to create a chart with chart I/Os

Introduction

You can add I/Os to a chart in order, for example, to:

- Insert the chart in a different chart and interconnect it with other charts or blocks
- Compile the chart as a block type

Procedure

Two methods are available:

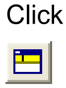
- Creating chart I/Os without assignment
- Creating chart I/Os with interconnection

Creating chart I/Os without assignment

In the first step, you create chart I/Os without reference to any parameters (for example, because the chart does not yet contain blocks and/or additional nested charts).
You then assign names, attributes and default values to the chart I/Os.
In the second step, you insert and interconnect the blocks/charts in the chart and assign the corresponding object I/Os from the chart to the chart I/Os.

Creating chart I/Os (1st step):

1. Click the button on the toolbar

   or
   select the **View > Chart I/Os** menu command.
   The dialog box for editing chart I/Os opens and is "docked" to the upper section of the chart window.
2. In the hierarchy window on the left, select the required I/O type (IN, OUT or INOUT).
3. In the detailed window on the right, edit the empty declaration line for the particular I/O type (name, data type, initial value, comment). Select the data type via a combo box.

Assign I/Os via drag-and-drop (2nd step):

1. Drag-and-drop the I/O of the block/chart to one of the chart I/Os that is of a compatible data type.

As an alternative with existing chart I/Os

You can assign the I/Os of blocks placed in the chart and/or nested charts to the existing chart I/Os without opening the chart I/O window.

1. Select the I/O, then select the menu command **Interconnection to Chart I/O...** in the shortcut menu or in the “Insert” menu.
   A dialog box opens that displays a list of all available I/Os of the relevant I/O type (available I/Os of IN and IN_OUT are displayed for IN under S7).
2. Select the required chart I/O and exit the dialog box by clicking “OK”.

   Note

   You can only assign unconnected I/Os with a compatible data type.

   You can also create interconnections the other way round. You drag a chart I/O to the required I/O of the block/chart with a compatible data type. Please observe the following information:

   – With outputs: The chart I/O must not yet be assigned to a block/chart I/O. However, a chart output that is not yet assigned can be assigned to the output of a block or chart that is already interconnected.

   – With inputs: You can assign the chart I/O to several block/chart inputs.

   If an I/O is already interconnected, it is not possible to change the data type.

Creating chart I/Os with interconnection

The first step is to create the chart itself. To do so, you insert and interconnect blocks/charts. In the second step, you open the chart I/O window and define the chart I/Os by connecting them to the I/Os of blocks/charts placed in the chart. A new line will always be created and all the properties of the connected I/O are adopted, for example, the name, attribute and initial value. Restrictions for the attributes S7_param and S7_link: If the values cannot be adopted, only the interconnection is created.
If naming conflicts occur, for example, because the same names are used in different blocks, the name is made unique in the chart I/O by incrementing it.

Create chart I/Os by connecting with <Ctrl> + drag-and-drop

1. Click the button on the toolbar

   or
   select the **View > Chart I/Os** menu command.
   The dialog box for editing chart I/Os opens and is "docked" to the upper section of the chart window.
2. In the hierarchy window on the left, click the required I/O type (IN, OUT or INOUT). The details window on the right displays the lines with I/Os. If new chart I/Os are yet to be created, these lines will be empty.

3. In the working window of the chart, select the required I/O on the block/chart, press and hold down the <Ctrl> key and drag the I/O to the “Name” box in the right window of the chart I/Os. The I/O is then adopted with its properties.
   Exception: interconnected I/Os are not reassigned.
4. Follow the same procedure for all other I/Os of the blocks/charts in the chart you want to interconnect with the chart inputs/outputs.

While holding down the <Ctrl> key, drag-and-drop an I/O that already exists in the chart I/Os to another empty line in the Chart I/Os window. A number is automatically appended to the name. The I/O name is then unique.

When you hold down the <Ctrl> key and drag-and-drop an internally interconnected I/O to a new line, a copy will be generated. The I/O will not be interconnected to the internal I/O.

Representation in the sheet bar

The CFC sheet bar displays the I/Os names and comments, I/O type and data type applied to the chart I/Os. The "interface I/O" type of interconnection is indicated by a small white triangle above the interconnection line.

Note

If an I/O interconnected to the chart interface has been hidden, there is no sheet bar entry. These interconnections can only be recognized in the object properties of the block ("Interconnection" column of the "I/Os" tab).

Changing Chart I/O Names

The chart I/O name does not need to include the name of the associated block I/O. You can rename it. If you want to rename it, select the name in the "Name" box and enter a new name. As an alternative, you can double-click on the start of the line of the chart I/Os in the right pane and enter the new name in the properties dialog.

System attributes

You can also assign system attributes to specific chart I/Os, as with block I/Os. The following rules apply:

When you interconnect an I/O of a block/chart already placed in the chart to a chart I/O:

- The chart I/O inherits the block/chart attribute, if an attribute has not yet been assigned.
- The chart I/O retains an attribute that has already been entered.
  Restrictions: For example, if text attributes are retained, with S7_link and S7_param an attempt is made to assign those values.
  Exception: The function will always adopt the value of S7_visible.

A chart with chart I/Os is not assigned system attributes, except the I/Os. These attributes can be assigned when you compile the chart as block type.

Assigning I/Os to inserted charts

You can also extend a chart with chart I/Os later by adding further chart I/Os. Adding I/Os may cause positioning conflicts if the target is a nested chart, i.e., a chart already inserted in another. In this case, as with overlapping blocks, the nested chart is displayed as an overlapping chart, in light gray and without I/Os.
Once the chart is positioned at a free location, the I/Os and interconnections are visible again.

You can find additional information on this in the following sections:

- Compile chart as block type (Page 361)
- Interconnection rules for chart I/Os (Page 155)

10.4 Special features of F-blocks in CFCs

Introduction

If "F-blocks" are used in a CFC, the following special features are available:

- Chart-based runtime group management for blocks of CFCs
  When F-blocks are also used in a CFC that is integrated in chart-based runtime group management, not only is a runtime group for blocks automatically created in the "standard program", but a runtime group for the included F-blocks as well, the "F-program".
  You can find additional information on this in the section "Chart-based runtime group management for blocks of CFCs (Page 192)".
- Password prompt of the safety program for CFCs with F-blocks
  This topic is described in the following section.

Password prompt for CFCs with F-blocks

The prompt for the password of the safety program (F-password prompt) for CFCs with fail-safe blocks (F-blocks) was made prior to version 8.2, for example, upon opening the CFC with the CFC editor.

You can find additional information on password prompting in the "Access protection" section of the "SIMATIC Industrial Software S7 F/FH Systems Configuring and Programming" manual.

The F-password prompt when the CFC is opened has been omitted in version 8.2.

As of version 8.2, the F-password prompt is only made for safety-related changes, for example, when changing the signature of the safety program. Therefore, changes to a standard program of a CFC with fail-safe components can now be carried out without requesting the F-password.

The F-password prompt is made regardless of whether the user creates an F-block explicitly or this occurs implicitly, for example, during a copy operation.

Applications for the F-password prompt

Copying a CFC with F-blocks:

- If you copy a CFC with F-blocks from source project "A" to the target project "B", then only the F-password of target project "B" is required.

Moving a CFC with F-blocks:

- If you move a CFC with F-blocks from source project "A" to the target project "B", then the F-password of target project "B" is first required and then the F-password of the source project.

Non-safety-related interconnections:

- Interconnections from F-blocks to standard blocks, converters for example, can be deleted without a password prompt. Creating such interconnections requires the F-password.
- The F-password is requested when creating and deleting the interconnections from standard blocks to F-blocks.

10.5 Handling block types

10.5.1 Creating block types in CFC

General

You can either purchase the block types you use in CFC in the form of block libraries or create them yourself. You can also expand the existing pool of blocks.

In CFC, block types can only be created by compiling CFC charts. If you want to create your own blocks with STL/SCL, you will find information in the "PCS 7 Blocks" programming guide.

A CFC chart can be compiled for further use (with chart I/Os) as block type.

You create a CFC chart, interconnect the blocks and create the chart I/Os for selected block I/Os.

Once the chart has been compiled, the block type is stored inside the block folder. By inserting it in the chart (from the S7 folder of the CFC block catalog) or by importing it using the menu command **Options > Block types...**, you can include the block type in the chart folder. In the block catalog, the block type is displayed under the family name specified for compilation.

The block type is managed via the object name FB xxx.

Make sure that the FB number is in the range supported by the CPUs on which you want the block to run. You can find information on these ranges in reference manual *Automation System S7-400, Module Data*. Example: Number of FBs = 512. The highest possible block number is thus FB 511.

You can assign a symbolic name to the block. This name will be entered automatically in the symbol table. The header name and symbolic name of PCS 7 blocks are identical.

Restrictions and characteristics

Before the CFC chart can be compiled as block type, it must satisfy the following conditions:

- The chart cannot contain nested charts, but may contain other block types created with CFC.
- Blocks can only be installed once manually in the run sequence, in other words, once outside the task assigned the "S7_tasklist" system attribute.
- All blocks have to be installed manually in the same task. All blocks of the chart to be compiled have to follow one another consecutively without gaps. There should be no blocks of other charts between them.
- The blocks cannot be installed in runtime groups.
- It is not possible to access the blocks in the chart externally, for example, using faceplates in WinCC. The blocks are initialized during runtime.
- The results (values) of the blocks in the chart are only applied to the outputs of the generated block type during runtime and after all blocks have been processed. Values are applied to the outputs even if ENO = FALSE is set.

- There cannot be any cross-connections with elements in the chart that is being compiled. Cross-connections of this kind are interconnections from another chart that bypass chart I/Os.
- The chart I/Os should not have an EN input since this is generated automatically.
- Signal inputs of the blocks have to be applied to the chart I/Os. Before signal inputs are assigned, the relevant chart I/Os have to be unhidden. The chart I/O inherits the block I/O attributes. Once the signal inputs have been assigned, the chart I/O can be hidden again.
- The interconnection of an ENO I/O defined at the chart I/Os is converted accordingly. If no ENO is defined, the ENO I/O of the block type provides the result of logical AND operations at the I/Os of all blocks called in the block type.
- The "S7_tasklist" system attribute represents the complete set of all task list attributes of the internal blocks.
- If a chart I/O is not connected to the I/O of a block contained in the chart, a warning is displayed during compilation.
  If one block I/O with the attribute S7_m_c is connected to a chart I/O, WinCC operations affect the same variable, regardless of whether the block I/O or the chart I/O was operated.
  In cases where several block I/Os are interconnected with a chart I/O, they will be provided with values during runtime. WinCC operations will be immediately overwritten, thereby rendering them ineffective. During compilation, this is recognized and a warning is displayed.

10.5.2 Overview of importing block types

Introduction

The following options are available to synchronize blocks / block types that are used in CFCs:

- Background import when inserting from the catalog to the chart.
- Import via the "local type update" in the CFC Editor:
  The changed blocks in the block folder of the project are made available to the CFC data management to allow you to use them in CFCs.
  You can find additional information about this in the description below.
- Central type update:
  You can use the process referred to as "central type update" to compare all blocks/block types and SFC types in the master data library and update them with those used in a project or the individual projects of a multiproject. If the project contains associated instances of the types concerned, they are also updated when the type update is performed centrally.
  You can find additional information about this in the sections "How to update block/SFC types in the multiproject (Page 110)" and "How to update block/SFC types in single projects (Page 112)".

Type update for CPU 410-5H PA:

CPU 410-5H PA supports type update in RUN mode. This makes it possible to update the instances and download them to the CPU in RUN mode after changing an interface at the block types.

You can find additional information on this in the section "Type update with a CPU 410-5H PA (Page 113)".

Note

Synchronization when using new channel blocks

When a channel block is imported from a newer library, sometimes additional blocks must be imported from the Basic Library and their interface needs to be changed.

This is why it is necessary to synchronize the blocks from the Basic Library before compiling your program in order to achieve a delta download in RUN.

In order to synchronize all the required blocks in this case, we recommend that you use the menu command "Options > Charts > Update block types" in the SIMATIC Manager and set the "Download in RUN" option.

Importing into CFC with the local type update

Blocks are made known to the CFC by their import. You can only use such blocks in CFC.

There are two ways to import blocks:

- By installing them in the chart from the catalog (import in the background)
- Import via the "local type update": via the menu command **Options > Block types...** in the CFC Editor.

Local type update is always useful when you require a lot of different blocks, for example, if you want to import a large number of blocks from the user program (block folder) to the chart folder. You would also use this dialog box if you have modified blocks (type modification) in the block folder that exist with the same name in the chart folder and need to be updated there. After this type of import, inserting from the catalog is much faster since import in the background is no longer necessary.

Tip: Imported blocks appear in the block catalog in the relevant block families and under "All blocks". You should then only insert blocks in the chart from this block catalog.

You can select the blocks you require in the "Offline block folder" box of the "Block types" dialog box and then import the blocks to the chart folder by dragging them with the mouse or clicking " -> ". If a block already exists in the chart folder, a warning is displayed. If an older version already exists, follow the steps outlined in the following section "Importing and using a new version of the block types (Page 107)".

Import via the SIMATIC Manager

Proceed as follows to perform an import via the SIMATIC Manager:

1. Open the corresponding library in the SIMATIC Manager.
2. Copy the block from the library block folder and paste it into the program block folder. Any existing block is overwritten.
3. Switch to CFC and select the menu command **Options > Block Types...**.

Handling blocks with identical numbers

Block numbers must be unique in an S7 program. The use of different block types (symbolic name and/or function) with the same object name (e.g., both are named "FB 61") is not allowed and will be rejected by the system. In this case, the block number must be changed.
Note: The object name consists of the block type "FB" or "FC" and the number.

During import, the system detects a block that already exists under this object name. An error message then appears in the "Block types" log. If the two blocks have different symbolic names, for example, the existing block name is "CTRL_PID" and that of the block you are importing is "CTRL_xx", the entry is made automatically in the symbol table. If two blocks have the same symbolic name, you have to make the entry in the symbol table yourself.

Make sure that the block number is in the range supported by your CPU. You can find more information on this:

- In the reference manual *Automation System S7-400; Module Data*
- With the menu command CPU > Module Information > "Performance Data" tab

The "No. of FBs" listed here is also the range for the block number. Example: Number of FBs = 512. The highest possible block number is thus FB 511.

10.5.3 Importing and using a new version of the block types

Introduction

If you want to use the modified types in CFC for previously used block types following a block type change, you need to import the modified block types into the chart folder. The block instances used in CFC are then adapted to the modified block types.

There are two options for the import:

- Import in the background, achieved by inserting a modified block type from the S7 program or the block library into the chart
- Import with the "local" type update using the menu command Options > Block Types in the "Block types" dialog box. Select the block to be imported from the list of source files ("Offline block folder", “Project”) and drag-and-drop it to the "Chart folder" list or click "→".

Using a new version

If a block type with this name but of a different version exists in the chart folder, the "Import New Version" dialog box appears displaying a corresponding message and version information on both blocks.

If you answer the "Do you really want to use the new version of the block types?" prompt with "Yes", all corresponding block instances are adapted. If you answer with "No", nothing is imported and no adaptation is made.

The following separate situations can be distinguished:

	Changes requiring neither a download of the entire program nor a new OS compilation because the changes are relevant only within the ES. Example: A block I/O was assigned the “hidden” attribute. In this case, you only need to download changes when the CPU is in RUN.
	Changes relevant to the OS and therefore requiring OS compilation. Example: After you have edited a message text.
	Changes in which the structure was modified. Example: By adding I/Os and/or messages. This can have the following consequences: • Interconnections and parameter assignments can be lost. • The only option is to download the entire program, in other words, the CPU must be switched to STOP mode. • If the block is intended for operator control and monitoring, in other words, it will be executed on the OS, you also need to compile the OS. • If you want to retain parameter settings from the AS, the chart must be read back before the next download.

Identical version

When you import block types of the same version as those in the chart folder, a dialog box displays the following:

- The message "The new versions of the block types from the offline block folder match the block types of the same name in the CFC. Importing is not necessary."
- A list of identical block types

You can find additional information about block changes in the section "Effects of type changes on block instances (Page 123)".

10.5.4 Inserting block types in CFCs

Block types in the catalog

You can drag-and-drop a selected block type from the libraries/block catalog to the chart. As an alternative, you can insert a block selected in the catalog at a free location in the chart partition in the overview or in the current sheet in the sheet view by pressing <Return>.

You can find additional information about searching for a block in the section "Searching for objects in the catalog (Page 445)".

Inserting block types

- Inserting from the libraries catalog
  Block types inserted from the libraries catalog will be implicitly imported into CFC.
  Block I/Os with modified system attributes
  If you attempt to insert a block in the chart from a library and when copying it to the block folder it is recognized that the system attributes differ from those of the existing block, the "Insert function block" dialog box opens.
  Click "Adjust attributes" to open the "Adjustment of system attributes" dialog box, where you can view all I/Os with different attributes.
  Click the attribute value "source" or "target (old)" to decide whether the I/O of the imported block [target (new)] will be assigned the attribute value of the "source" or the "old" target.
  Click “OK” to overwrite previously used attributes with the current values. Both dialog boxes are closed.
- **Inserting from the block catalog**
  Blocks from the S7 program are automatically imported to CFC when you insert them in the chart and also appear in a family of CFC blocks. The block icon appears in different forms, depending on whether the block is already known to CFC, in other words, whether or not it has been imported.
  Blocks that were not imported are indicated by the following icon:

  Blocks in the S7 program are still unknown in CFC, even if a block of this type has already been imported. This means that when you want to insert a block from the S7 program into the chart, a check is always made to determine whether the block has already been imported.
  You can find blocks that are already imported in the "All blocks" directory and in one of the block families. A block that is not assigned to a family (indicated in the header) can be found in the "Other blocks" directory. These blocks are indicated by the following icon:

  Tip: Inserting blocks from the block families is the fastest way to position blocks in the chart, since these are already known in the CFC and no longer need to be verified.
  You should always select this method if a block type has already been imported.

Note on startup blocks

Note the following information on startup blocks, for example, controller blocks CONT_C, CONT_S or PULSEGEN, relevant to specific CPUs:

- With S7-3xx and S7-4xx CPUs, the block must not only be called in the cyclic interrupt OB, for example, OB 35, but also in OB 100 (startup OB) so that their startup code can be processed. The block is installed automatically.
- With the S7-3xx CPU, note that the RESTART block (FC 70) must be the first block called in OB 100. Each time this block is called, memory word MW 0 is changed. This allows blocks called in OB 100 to recognize the startup. MW 0 must not be used elsewhere in the CPU.

Blocks with identical numbers

Object names (block type + block number) must be unique in an S7 program. You can find additional information about this under "Handling blocks with identical numbers" in the section "Overview of importing block types (Page 105)".

10.5.5 How to update block/SFC types in the multiproject

Introduction

In the multiproject, you can use the so-called central type update to compare all block and SFC types used in the individual projects with those of the master data library and update them accordingly. At the same time, you can also update the templates in the chart folder of the master data library.

Assuming that the latest versions are always kept in the master data library, you can use the update function to make sure that all or only certain types have the same current version in all or selected projects of the multiproject. If the project contains associated instances of the types concerned, they are also updated when the type update is performed centrally.

Note

"Type update in RUN" with CPU 410-5H PA

CPU 410-5H PA supports type update in RUN mode. This makes it possible to update the instances and download them to the CPU in RUN mode after changing an interface at the block types.

This is why the procedure for a CPU 410-5H PA has changed.

You can find more information on type update in RUN in the section "Type update with a CPU 410-5H PA (Page 113)".

Procedure

Proceed as follows to update block/SFC types in the multiproject:

1. Select the block folder or chart folder of the master data library in the SIMATIC Manager or the blocks or SFCs you want to update in them.
2. Select the menu command **Options > Charts > Update block types...**.
   The dialog page 1(3) "Select the S7 programs to be checked" opens. All existing S7 programs in the multiproject are listed there and already selected.
   The "Download in RUN" column displays a check box when the following condition is met:
   – The program is in a CPU 410-5H PA.
   – Changes can be downloaded for this program, which means there is no need to download the complete program.

   By selecting the check box, a type update in RUN can be performed for this program (possible only in a CPU 410-5H PA). In addition, the "Calculate..." button is displayed in the "Resources" column.
3. Check the selection and, if required, exclude the S7 programs in which you do not want to update any blocks/SFC types.
4. Click "Next".
   A check mark in the "Status" column shows the programs already checked.
   When the check is completed, dialog page 2(3) opens. On the "Select the block types or SFC types to be updated." page, you can see the types selected in the library that have a different version in the individual S7 programs.
   To help you decide whether to select blocks, the "Consequence" column shows you the effects of the new import. The following consequences are displayed:
   – **All associated instances will be modified** (no additional consequences).
   – **The OS must be compiled again.**
     All associated instances are adapted. You can then launch the wizard to compile the OS with the menu command **Options > Wizard: "Compile Multiple OSs" > Start...** or by selecting an OS and using the shortcut menu command Compile.
   – **Loading entire program in operating mode STOP**
     After compiling, you can no longer download changes in RUN mode. All associated instances are adapted.
     Read the information provided in the introduction when using a CPU 410-5H PA.

   Note: When updating SFC types, the button "Show differences" is shown on this page 2(3). When the "Version Cross Manager (VXM)" add-on package is installed, you can use this button to open VXM and display the detailed information about the differences.
5. Check the selection and, if required, exclude those objects you do not want to update due to the consequences.
6. Click "Finish".
   A check mark in the "Status" column shows which objects have already been updated.

7. When the update is completed, you move on to the next dialog box page 3(3), "Log". Here, you can see all the information on what took place during the update and the results. Examples:
   – Start of type update
   – The log path
   – The library
   – The selected objects
   – The selected S7 programs
   – The updated objects
   – Deleted interconnections
   – End of type update

   Note

   Check the entries in the log. The central type import can have an effect in the instances which requires an action, for example, interconnections may have been deleted that must be checked in the instances and adapted, if necessary.

8. If you want to print the log, click "Print", otherwise click "Close". The dialog box closes.

If you stop the update with "Cancel", all the updates made up to this point are retained. These are also included in the log.

You can find additional information on this in the section "How to update block/SFC types in single projects (Page 112)".

See also

Special considerations during download to a CPU 410-5H PA (Page 386)

10.5.6 How to update block/SFC types in single projects

Introduction

You can update block/SFC types of a library that is not the master data library of a multiproject in individual projects of your choice. If the project contains associated instances of the types concerned, they are also updated when the type import is performed centrally.

Procedure

Proceed as follows to update types in a single project:

1. Open the required library in the SIMATIC Manager, for example, PCS 7 Library.
2. Select the block/SFC types you want to update in a project in the block/chart folder of the library.

3. Select the menu command **Options > Charts > Update block types...**.
   The "Open project" dialog box opens.
4. Select the project in which you want to update the types and click "OK".
   Page 1(3) of the dialog box "Update block types" opens.
5. Follow the steps described in the section: How to update block/SFC types in the multiproject (Page 110) (starting at point 3).

10.5.7 Type update with a CPU 410-5H PA

Overview

CPU 410-5H Process Automation (CPU 410-5H PA) supports type update in RUN mode.

This makes it possible to update the instances and download them to the CPU in RUN mode after changing an interface at the block types. This is only possible in STOP mode in other automation systems.

The update with the "Local" and "Central" type update is described in the paragraphs below.

The abbreviation "TCiR" is also used for this function. This stands for "Type Change in RUN".

Definition of an interface change

There is an interface change when at least one of the following conditions is met:

- A data type of an I/O has been changed.
- An I/O has been added, moved and/or deleted.
- The default value of an I/O has been changed.
- An I/O with message response has been renamed (can only be loaded in STOP mode)

Advantages

- With a CPU 410-5H PA , types can be updated after an interface change and downloaded to the CPU in RUN.
- The previous limit of 56 simultaneously linkable blocks in the CPU does not apply with CPU 410-5H PA .

Note on name change in the interface

- I/O without message response
  - If an I/O name is changed at a block type and another interface change is also carried out, for example, an I/O is added, the name change will not be recognized as such.
  - The I/O with another name is then interpreted as a new I/O and thus assigned the default value from the block type. The process value is lost.
- I/O with message response
  - Can only be downloaded in STOP mode

Maintaining the capability for type updating in RUN

Type updating in RUN is no longer possible if the following change has been made at a block:

- Name of an input with configured message was changed ("Message Event ID").

Note

Changing associated values

If, for example, you call an "ALARM_8P" or "NOTIFY_8P" block in a block and wish to use additional associated values in it, a type update in RUN is possible, but this extension only takes effect after a restart of the CPU for system-related reasons.

If the associated values SD_1 to SD_5 are assigned, all associated values up to SD_10 are assigned.

Extending a block with initialization-relevant parts

The following must be considered when extending a block with parts that are needed for an initialization:

Downloading the change via the type update in RUN can be carried out without a startup being performed and thereby initialize the extension.

To prevent this, the required initialization code must be implemented not only in the startup sequence (OB100), but also separate from the sequence under "Initial run bit" in a new sequence that only initializes the newly added part.
This requires a new date comparable to the "Initial run bit" to be added. This triggers the execution of the initialization sequence for the initial run after the type update in RUN.

If initialization-relevant extensions are again made to the block with an additional change, this must be reflected in another initialization sequence, which means the block will contain 3 initialization sequences.

An example of an initialization-relevant extension is the addition of an "ALARM_8P" instance to the block. This must be called at the initial run of the block.

Default value change on interfaces of blocks / block types

In the case of default value changes on the interface of a block/block type, this block could not be downloaded with the type update in RUN, but only in STOP mode of the CPU. As of CFC V8.2, it is also possible to download a module with such a change via the type update in RUN.

The user can choose between the following options for the type update in RUN with changed default values:

- All previous default values are retained at all existing instances of the block.
- All new default values are applied to all existing instances of the block.

You can select between the "local" or the "central" type update:

- Local type update in the "Import new versions - Download changes in RUN" dialog box
  In this dialog box, the option buttons "The previous default values will be retained" and "Activate new default values" are offered for selection.
  The option buttons are only available when the "Download of changes to the CPU in RUN mode" check box is selected.
  Depending on the selected option, the changed default values are applied to all instances of the blocks or the previous default values are retained.
  The setting applies to all blocks / block types with changed default values. If a different reaction is desired for different blocks/block types, the procedure for the local type update for each modified block/block type must be performed separately.
  The procedure is described in the paragraph below.
- Central type update in the "Update block types" dialog box
  You can use the process referred to as type update to compare all block types and SFC types used in the individual projects with those of the library and update them accordingly.
  The "Update block types" dialog box displays all changed blocks/block types and you can determine for each block/block type per S7 program whether the changed default values are to be applied. The selection is made via the associated check box in the "New default values" column.
  The procedure is described in the paragraph below.

The following restrictions exist for applying modified default values:

- Only all the modified default values of a block type can be applied to the instances. It is not possible to select only some modified default values.
- A modified default value can only be applied to all instances of the block type. It is not possible to apply the modified default value only to certain instances.

Note

Effect of applying default values during download to the CPU

When applying default values, all default values are always applied during the local and central type update.

This means that values that were changed by the process are overwritten as well during download to the CPU.

Requirements

- The target system contains a CPU 410-5H PA and is accessible online.
- The CPU 410-5H PA firmware has version V8.1 or higher.
  If this firmware version is not in place, the corresponding option is not available in the dialog box for loading to the target system. In this case, only a complete download is possible.
- For migrated projects, the "Type update in RUN" function is only possible if a complete download or download of changes has been performed at least once with CFC version 8.1 (or higher) before block types are changed and before download to the CPU. Only then can the "Download of changes to CPU in RUN mode" check box be operated for migrated projects.

Principle of a type update in RUN with local type update

Here, the basic procedure for the local type update is described. The procedure for central type update is described in the following paragraph.

1. If blocks in the block folder of the project have been changed, they can be imported using the process referred to as local type update into the CFC data management and synchronized.
   A CFC of the project is opened in the CFC Editor for this.
2. The "Block types" dialog is opened with the menu command **Options > Block Types...**. Then, a block is selected in the "Chart folder" list and the "New Version ..." button is now operable.
   If the block has been changed, clicking the button opens the "Import new versions - ..." opens the dialog box.
   If the selected block has not been changed, this is displayed in a dialog box and the type update can be canceled or continued with another block.
3. For S7 programs in a CPU 410-5H PA , the "Import new versions - Download changes in RUN" dialog box is displayed with an additional "Download of changes to the CPU in RUN mode" check box. Use this option to activate downloading in RUN.
   If the "Download of changes to the CPU in RUN mode" check box is selected, the "The previous default values will be retained" and "Activate new default values" option buttons are operable.
   These option buttons are used to specify how the default values are to be handled after changing default values on interfaces of blocks / block types with "Type update in RUN".
   You can find additional information on this in the paragraph above.

4. If the "Download of changes to CPU in RUN mode" option is enabled, a status indicator ("!" symbol and supplementary text) appears in the status bar of the CFC editor after closing the dialog box to inform you that a download to the CPU is still required.
   This display is practical for the user, because it may not be possible to perform a download immediately after the type update. Instead, the configuration may need to be tested and completed, for example, because textual interconnections may be incomplete due to the type update or new interface parameters are not yet interconnected.
   Additional configurations, for example, even another type update, can be performed directly after the type update and before the next download. However, the capability to download in RUN can be lost due to these subsequent configuration actions. The download should therefore be performed as soon as possible after a type update with the "Download of changes to CPU in RUN mode" option enabled.

   Note

   Multiple type updates prior to download

   If several type updates are carried out prior to downloading and download in RUN mode should remain possible, the "Download of changes to CPU in RUN mode" option must be enabled for each type update. This means if one of the previous type updates was performed with this option disabled, a subsequent download is no longer possible in RUN.

5. The program is subsequently compiled and downloaded as usual.
   If download in RUN is possible a corresponding option is available in the dialog box. If this option is not available, check the requirements described above.
   For interface changes, a calculation is made to determine whether it is possible to adapt all changed instances in one operation. If the capacity of the CPU is exceeded by this calculation, a corresponding message appears and the user can change or reduce the number of types to be updated.
   If CPU capacity is exceeded even after the reduction of the number, the type update must be performed in STOP mode as usual.

   Note

   Access of the OS to blocks with a changed interface

   After download, attempts by the OS to access blocks with a changed interface are not possible in this state and the affected picture objects are disabled, not displayed at all or cannot be operated. This state remains until the OS has been compiled and downloaded.

   However, all automation systems operated and monitored from an OS do not need to be updated first before compiling and downloading of the OS. This is possible because structure types are formed AS-specific and can handle different block types.

Principle of a type update in RUN with central type update

1. Select the block folder or chart folder of the master data library in the SIMATIC Manager or the blocks or SFCs you want to update in them.
2. Select the menu command **Option > Charts > Update block types...** .
   The dialog page 1(3) "Select the S7 programs to be checked" opens. All existing S7 programs in the multiproject are listed there and already highlighted as selected.
   The "Download in RUN" column displays a check box when the following condition is met:
   – The program is in a CPU 410-5H PA.
   – Changes can be downloaded for this program, which means there is no need to download the complete program.

   When the option is enabled in the Download in RUN column, the "Calculate..." button opens a dialog with information about the resources that are needed for downloading in RUN. Information about the resources required can then be compared with the resources currently available in the CPU and, if necessary, to change the selection of block types to be updated and downloaded.
3. Check the selection and, if required, exclude the S7 programs in which you do not want to update any blocks/SFC types.
   Click "Next".
4. A check mark in the "Status" column indicates which programs have already been checked. When the check is completed, dialog page 2(3) opens.
   On the "Select the block types or SFC types to be updated." page, you can see the types selected in the library that have a different version in the individual S7 programs.
   The affected block type is displayed in the "Block type" column. Now you can select the block types that are to be updated.
   In the "New default values" column, you can determine whether changed default values of the interface blocks are also to be transferred to the instances of the blocks. Select the appropriate check boxes of a row to apply the changed default values to all instances of the block.
   The effect of the type update appears in the "Consequence" column as a decision aid for selecting. The following consequences are displayed:
   – **All associated instances will be modified** (no additional consequences).
   – **The OS must be compiled again.**
     All associated instances are adapted. You can then launch the wizard to compile the OS with the menu command **Options > Wizard: "Compile Multiple OSs" > Start...** or by selecting an OS and using the shortcut menu command Compile.
   – **Loading entire program in operating mode STOP**
     After compiling, you can no longer download changes in RUN mode. All associated instances are adapted.
   – **Download in RUN is possible after adapting all instances.**
     Once all associated instances have been adapted and the program is compiled, a download of changes in RUN mode is possible.
     This is only possible when using CPU 410-5H PA.

   Note: When updating SFC types, the button "Show differences" is shown on this page 2(3). When the "Version Cross Manager (VXM)" add-on package is installed, you can use this button to open VXM and display the detailed information about the differences.

5. Check the selection and, if required, exclude those objects you do not want to update due to the consequences.
6. Click "Finish".
   A check mark in the "Status" column shows which objects have already been updated. When the update is completed, you move on to the next dialog box page 3(3), "Log".
7. Proceed with step 7 in the section "How to update block/SFC types in the multiproject (Page 110)".

See also

Special considerations during download to a CPU 410-5H PA (Page 386)

10.5.8 How to delete block types

Deleting block types from the chart folder

Call menu command **Options > Block Types...** to open a dialog box where you can delete block types that are no longer required from the chart folder and, if necessary, also from the block folder. Select the blocks and click "Clean up" to delete the blocks from the list.

Clean Up

Click "Clean up" in the "Block types" dialog box to open two dialog boxes in succession in order to delete blocks from the chart/block folders.

1. "Clean up blocks in CFC" dialog box
   This lists all block types contained in the chart folder but not used in the charts. In other words, there are no instance blocks.
2. "Clean up blocks in block folder" dialog box
   This lists all block types contained in the offline chart folder but not used in the charts (the block types are not in the chart folder).
   These could also be block types that are called from other blocks und are normally only found in the block folder.

Example:

The CTRL_PID (FB 61) block, which calls the OP_A_LIM (FB 46) block, was deleted in the chart folder, but the OP_A_LIM block is still in the block folder.

Select the corresponding blocks in each dialog box and delete them from the respective target (chart/block folders) with "OK".

10.6 Editing blocks

10.6.1 Editing blocks

Overview

The following sections contain information about editing blocks:

10.6.2 Inserting blocks

10.6.2.1 Inserting blocks

To insert a block means that you select a block type and insert it in the chart. You can insert the block easily in different ways:

- Drag-and-drop it from the block or library catalog;
- If the block is selected, use the <Return> key.

When a block is inserted, it is assigned a name that is unique within the chart.

The inserted block is an instance of the block type. You can create any number of block instances from each block type.

Note

The block type comment is not transferred for the block instance.

Each inserted block is assigned default runtime properties, which the user can modify.

You can find more information about inserting blocks in:

Catalog of blocks, charts, templates and libraries (Page 48)

Searching for objects in the catalog (Page 445)

Settings for inserting blocks (Page 121)

Positioning large blocks (Page 121)

Inserting unplaced blocks (Page 122)

10.6.2.2 Settings for inserting blocks

Default runtime properties

When inserting a block, you must also specify its runtime properties, for example, the block insertion position in the run sequence.

As default:
The block is inserted following the block that is displayed in the status bar.

The status bar displays one of the following:

- When you first create a chart, the default of the specific PLC
- The last block to be inserted
- The block specified by the run sequence

The current insert point is displayed to the right in the status bar. It displays the following:

- Task name (OB x)
- Chart name
- Name of the block which will serve as the predecessor in the run sequence the next time a block is inserted in the CFC chart

Additional information

You can find additional information on this in the following section: Runtime properties (Page 174)

10.6.2.3 Positioning large blocks

Introduction

You can also insert large blocks, in other words, blocks containing a high number of visible block I/Os. To do this, position the block across the horizontal dividing line of two or three successive sheets. In this case, a copy of the first block header is inserted immediately below the dividing line.

Notes

- If a positioning conflict occurs, the block is shown at this position as an overlapping block, light gray and without I/Os. A positioning conflict occurs when a block is inserted at the sheet boundary, with the result that only the header can be displayed in the first sheet and not all I/Os (to avoid a double header). You can change this layout by moving the block slightly in the vertical direction.
  The same applies to "small blocks", in other words, blocks that do not have a body with designated I/Os such as "AND" and "NOR".
- If the block is too large for those three sheets, the I/Os that cannot be displayed will be hidden automatically. The maximum number of visible I/Os amounts to 160 inputs and 160 outputs. You can access all I/Os using the menu command **Edit > Object Properties...**.

Tip: In order to improve your overview, it is useful to show only the I/Os you are going to use for interconnections or monitoring in test mode.

10.6.2.4 Inserting unplaced blocks

Unplaced blocks in the catalog

You can take unplaced blocks from the "unplaced blocks" catalog and position them in the chart again. To do this, click the button

Note

This catalog is only available if unplaced blocks actually exist.

The block name and the complete path (chart name) are specified. This allows you to see the block source. Use the shortcut menu command **Open Original Chart** to open the chart from which the block originated.

You can also insert unplaced blocks from a different chart in the active chart. Previous interconnections to blocks or to the sheet bar are not regenerated when the block is positioned in the chart again.

10.6.3 Global block type change

10.6.3.1 What you should know about global block type changes

What is a global block type change?

When you modify the interface description and/or system attributes of a block type and import it to the CFC data management, an existing block type of the same name can be replaced and, therefore, updated, by this new version. All block instances of this type will be updated to the new block type.

The ability to modify the global block type relates to FBs and FCs. BOP types cannot be changed since they are an integral component of CFC.

Before the changes are applied, a warning is displayed, pointing out the effects and showing data of the old and new block type, such as the name and date of last interface modification. At this point, you can cancel the update by clicking “No” or accept the changes by clicking “Yes”.

Global modifications of the type may produce unwanted effects on the block instances. For example, interconnections and parameter settings may be lost. In this case, you must modify the block instances manually.

Modifications that result from a global type change are logged and automatically displayed following the update. You can also display this log using the menu command **Options > Logs: Block Types...**. If a modification of block instances is required, this log helps you save time while also reducing the risk of errors.

Situations preventing block type change

If the category of the block type has been changed (for example, FB <--> FC), the block type is not changed and a corresponding message is written to the log.

Additional information

You can find additional information about this in the following sections:

Effects of type changes on block instances (Page 123)

Tolerant type import after data type extension (Page 125)

10.6.3.2 Effects of type changes on block instances

Introduction

In CFC, when you modify the interface description (block I/Os) or the system attributes of a block type, all instance blocks are automatically adapted. You should take into account any relevant effects of an interface modification.

Block comments are changed globally in the block instances (overwritten) following a block type change, regardless of any previous instance-specific modification.

Modification of block I/Os

Modifications to block I/Os affect the instances as follows:

- I/O was added:
  The instances will be updated accordingly. The default system attributes will be set. If no more space is available at this position after you have increased the block size, the block will be displayed as an overlapping block.
  For PCS 7, you can find information below in the section "Effects on WinCC".
- I/O was deleted:
  The I/O is removed in the instances. If this is an interconnected I/O, the interconnection or an SFC access will also be deleted. The deleted interconnection or SFC access is entered in the log of the changes.
  For PCS 7, you can find information below in the section “Effects on WinCC”.
- I/O data type was modified:
  Result of deleting an I/O and of the creation of a new I/O.
  For PCS 7, you can find information below in the section “Effects on WinCC”.
- An I/O was renamed:
  The reference to the old name cannot be generated automatically by the system. Thus, this is the result of deleting an I/O and of the creation of a new I/O.
  For PCS 7, you can find information below in the section “Effects on WinCC”.
- The order of the I/Os was changed:
  The order of the I/Os is taken into account. The interconnection, parameter assignments and attributes will be retained.

Effects on values, comments and system attributes of the block I/Os

General guideline on the effects of changing attributes: What cannot be changed for a specific instance is updated automatically when the type is changed.

Values, comments and system attributes S7_string_0, S7_string_1, S7_unit, S7_shortcut can only be changed globally at the I/Os of block instances if no instance-specific changes were effected previously. This means that only the default type-specific values, comments and system attributes will be changed.

If the system attribute "S7_archive" of a block type is changed, this attribute change is not made to existing instance blocks following the type import.

Effects on WinCC (PCS 7)

When you modify block type data, DBs with new DB numbers can be created by compressing and compiling. In order to maintain the possibility of online access, the data must be transferred again to WinCC (OS compilation).

If this change affects block I/Os intended for operator control and monitoring (attribute S7_m_c=true), the following rules apply:

- An I/O added will be made known and used in WinCC following the OS compilation.
- After an I/O is deleted, the WinCC tag no longer exists. Any existing interconnections must also be deleted in WinCC.
- When the name of an I/O is changed, the name of the WinCC tag also changes. Interconnections of picture elements, block icons and faceplates must be adapted.

As a general rule: All changes affecting WinCC are followed by a new OS compilation.

Note

Changes to block I/Os have an effect on compilations and downloads. Following an interface modification, only compilation of the entire program is possible. If there is already an older version of the modified block type loaded, the entire program must also be downloaded.

10.6.3.3 Tolerant type import after data type extension

Data types

If the data type for a block type is extended, for example, from INT to DINT, all I/O data will be retained during a global type change if the old data type can be mapped on the new one without conversion. This applies to the following data types:

BYTE	→	WORD
BYTE	→	DWORD
WORD	→	DWORD
INT	→	DINT
STRING1n	→	STRINGn+m (e.g. STRING16 → STRING32)

Note

Remember that interconnections to these I/Os will still be lost due to the global type change.

10.6.4 Specifying object properties

10.6.4.1 Specifying object properties

The editing of object properties is described in the following sections:

How to enter block names (Page 126)

How to enter comments (Page 126)

How to assign block icons to specific instances (Page 127)

How to specify the number of I/Os (Page 127)

10.6.4.2 How to enter block names

Introduction

A name is assigned to the block when you insert it. You can modify this name. The block name must be unique within the chart. The CFC verifies this uniqueness. The name has a maximum length of 16 characters and is displayed in the block header. Prohibited characters: / \ . " %.

Note

CFC in PCS 7:

Remember when assigning names that the variable name must not be longer than 128 characters for transfer to the OS. This name consists of the following components:

- The name of the folder in the hierarchy path
- The chart name
- The block name
- The separator (period)
- The I/O name (variable name)

Editing block names

To edit block names, proceed as follows:

1. Select one or several blocks
   and then select the **Edit > Object Properties...** menu command.
2. Select the "General" tab.
3. Enter the required name in the "Name" input field.
4. Confirm your entries by clicking “OK”.

The dialog box is closed and, if multiple blocks were selected, the dialog box of the next block appears.

10.6.4.3 How to enter comments

Introduction

The comment, a user-specific text, appears in the block header. The block header can display up to 14 characters of comment. Small blocks do not display the comment.

Procedure

To enter block comments, proceed as follows:

1. Select one or several blocks
   and then select the **Edit > Object Properties...** menu command.
2. Select the "General" tab.
3. Enter the required text in the "Comment" input field.
4. Confirm your entries with "OK".

The dialog box is closed and, if multiple blocks were selected, the dialog box of the next block appears.

10.6.4.4 How to assign block icons to specific instances

Block icons

Blocks intended for operator control and monitoring can be displayed under WinCC by a block icon (for example, used for calling a faceplate).

Different block icons available for a block type can be assigned to specific instances (for example, to indicate different type versions: MOTOR block as motor, fan or pump).

In the “Special object properties” group, specify the block icon displayed under WinCC in the “Block icon” input field, using a maximum of 16 characters.

This input field is only enabled if you have selected the "OCM possible" check box.

10.6.4.5 How to specify the number of I/Os

Introduction

You can add or delete inputs at blocks that have a variable number of inputs of the same data type (for example, NAND or OR).

Procedure

To change the number of inputs:

1. Select the corresponding block and then select
   the **Edit > Number of I/Os...** menu command.
   The dialog box of the same name opens.
2. Enter the number of inputs (2 to 120) in the input field and click "OK".

   Note

   If the available space is insufficient for the new block length, the block will be displayed as an overlapping block that you can move within the chart.

   You can only reduce the number of I/Os down to the lowest interconnected I/O.

10.6.5 Copying blocks

10.6.5.1 What you should know about copying blocks

Introduction

You can copy blocks and nested charts both within charts or to other charts. It is also possible to copy several or all the objects of a chart at the same time. This allows you to duplicate tested substructures quickly and without errors.

Copying blocks/nested charts

When copying blocks/nested charts, you must consider the following effects on interconnections:

- Interconnections between objects copied in groups will be retained.
- Interconnections between copied objects and external elements will be deleted.
- Interconnections to shared addresses are either copied also or removed, depending on the setting in the "Settings for Copying/Moving" dialog box. You can display this dialog box using the menu command **Options > Customize > Copy/Move...**.
- Names are retained if possible. A corresponding error number will be appended to the names if there is a conflict.

Runtime properties of copied blocks

When you copy blocks, the response is basically the same as if you were adding blocks from the catalog:

- All copied blocks will be inserted at the position specified by the currently set "Predecessor for Insertion Position". Multiple instances of blocks in a cyclic task are transferred and inserted at the end of the OBs.
- Blocks the CFC will also automatically insert in different tasks (for example, blocks with startup relevance) will be inserted according to their task list.
- The runtime groups will not be copied. Blocks of those groups are installed in their previous order according to the block installation pointer of the destination.

Changed behavior with chart-based runtime group management (as of V8.2):

When blocks are copied to a CFC with activated chart-based runtime group management, the blocks are always inserted at the end of the runtime group regardless of the current position of the insertion pointer.

Checking the block types and the system attributes

When blocks are copied, the block types and the system attributes are compared with the existing block types and system attributes in the destination. The check is performed according to the following rules:

Block type

- Check the system attributes of the block type.
- Check the availability of all I/O types.
- Check the identity of the following names/types:
  - I/O type name (for example, MODE)
  - I/O type (for example, IN_OUT)
  - Data type (for example, DWORD)
  - If data type STRING: Check string length.
  - If data type STRUCT: Check data type.
- Check the system attributes of the I/Os.
- Check if the block types (SFC type, other type) are the same in the source and the destination.

System attributes of the block type

- Check if the system attributes s7_alarm and s7_m_c, which are always available, match.

System attributes of the block I/Os

- Check if the system attributes s7_a_type, s7_m_c, s7_link, and s7_param, which are always available, match.
- Check if the optional system attributes s7_server, s7_ts and s7_qc are available and if they match.
  - If they are in the source, they must be available in the destination.
  - If they are not in the source, they cannot be available in the destination.
  - If they are in the source and the destination, they must have the same values.
- Check if the optional system attributes s7_shortcut, s7_unit, s7_string0, and s7_string1 are available.
  - If they are in the source, they must be available in the destination.
- The optional system attribute s7_enum is always included when copying within a project. If the block is inserted in a different project that does not contain the enumeration in the shared declarations, the system attribute is inserted as a numeric value.

Additional information

You can find additional information on this in the following sections:

How to copy blocks (Page 130)

How to move blocks (Page 131)

10.6.5.2 How to copy blocks

Requirements

- There must be enough space in the target chart to paste the copied blocks.
- The relevant charts must be open.
- The charts must be located on the same CPU.
- If you want to copy blocks to other sheets, select the overview or open another chart window that displays the destination sheet.
  - You can find additional on the overview in the section: Views in CFC (Page 82)

Copying blocks to other sheets or charts

You have two copy options:

Drag-and-drop

1. Select the blocks you want to copy.
2. Hold down the <Ctrl> key and the mouse button to drag-and-drop one of the selected blocks to the new position in the same or a different chart.

Copying/pasting

1. Select the blocks you want to copy.
2. Click the button for **Edit > Copy**

   The selected blocks will be highlighted.
3. Activate the chart to which you want to copy the blocks and click the **Edit > Paste** button

   The blocks are inserted in the other chart at the same position, offset by one grid section to the right and down. If this position is already occupied by other objects, the blocks are displayed as overlapping blocks. Otherwise, they are displayed as usual.
4. In this case, you can move the blocks to an appropriate position. The AS can also execute overlapping blocks.

Copying blocks to other CPUs

You can copy blocks from your CPU to external locations. This operation affects functionality in the same way as operations for copying charts to a different CPU.
You can find additional information on the insertion position in the run sequence in the section: What you should know about copying blocks (Page 128)

10.6.6 Moving blocks

10.6.6.1 How to move blocks

Introduction

You can move blocks to other chart positions and charts. It is also possible to move block groups or all blocks at once.

Moving blocks has no effect whatsoever on the functionality. This means that the interconnections and runtime properties remain the same. Block names are retained if possible. A corresponding error number will be appended to the names if there is a conflict.

Moving blocks within a chart

You move one or more blocks within a chart as follows:

1. Select the block(s).
2. Drag-and-drop the selected block(s) to the new position in the chart.

In the overview, you can also move blocks to other sheets.

Note

"Cut" and "Paste" operations are not possible within the same chart partition. If you move a block to another chart partition, the object is inserted in the same sheet and at the same position it had in its original chart partition.

If objects are already placed at the insertion position and the blocks overlap after insertion, they are displayed in a different color. The color can be configured in the "Color settings" dialog using the "Overlapping blocks" option.

If blocks do not overlap, they are displayed as usual. In this case, you can move the blocks to an appropriate position. The AS can also execute overlapping blocks.

Moving blocks to other charts

Requirements

The chart to which you want to move blocks must be open.
The charts must be located on the same CPU.

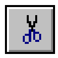
You have two options for moving blocks to other charts:

Drag-and-drop

1. Select the blocks you want to move.
2. Drag-and-drop the blocks to the new position in the other chart.

Cut and paste

1. Select the blocks to be moved and click the **Edit > Cut** button.

   The selected blocks are highlighted in light gray.

2. Activate the chart to which you want to move the blocks and click the **Edit > Paste** button.

   The blocks are inserted in the other chart at the same position, offset by one grid section to the right and down. "At the same position" in this context means that the block is also inserted in the sheet with the same sheet number.
   If this position is already occupied by other objects, the blocks are displayed as overlapping blocks; otherwise they are displayed as usual. In this case, you can move the blocks to an appropriate position. The AS can also execute overlapping blocks.

   Note

   You cancel the cut function when you initiate a function other than **Paste** after you have selected the menu command **Edit > Cut**. The blocks you cut will remain at their original position.

Effects of moving

Moving blocks has no effect whatsoever on the functionality. This means that the interconnections and runtime properties remain the same. Block names are retained if possible. A corresponding error number will be appended to the names if there is a conflict.

10.6.7 Deleting blocks

Deleting blocks

You can delete one or multiple blocks via the <Del> key or the menu command **Edit > Delete**.

Deleting blocks from the chart has the following effects:

Effects on interconnections

When you delete blocks, all interconnections between blocks to be deleted and objects you have not deleted will be lost.

If a block output is interconnected to a block that is not to be deleted, a warning is displayed. You can then decide whether to continue and delete the block. Instead of the truncated interconnections, the block inputs of the block that is not deleted are assigned default parameter values.

Effects on runtime properties

The following rules apply to the default insertion position in runtime groups:

- When you select to delete the "Predecessor for Insertion Position" block, the preceding object inherits this attribute.
- After you have deleted all blocks in a runtime group, the installation pointer moves to the start of this runtime group.

Additional information

You can find additional information about this in the following sections:

How to delete block types (Page 119)

Runtime properties (Page 174)

10.6.8 Aligning blocks

10.6.8.1 How to align blocks

Introduction

The system can align blocks you have inserted in a chart and blocks already assigned interconnections.

Procedure

1. Select the blocks you want to align.
2. Select one of the menu commands **Edit > Align > Left/Right/Top/Bottom**.
   All blocks selected are aligned to the left/right/top/bottom.

Result

The blocks are aligned at the visible edge of the block graphic, not at the frame of the selected blocks. The block to the extreme left or right or nearest to the top or bottom determines the alignment position for the blocks.

If alignment causes a conflict, for example, overlapping blocks, an error message is generated and the old layout will be restored. The blocks will not be aligned in this case, not even those that do not cause a conflict.

Note

This align command is disabled when blocks selected for alignment include an overlapping block (represented in another color).

10.6.9 Editing I/Os

10.6.9.1 How to set I/O parameters

Definition

In this context, setting parameters refers to the assignment of parameter values and, if required, attributes to a block or chart I/O. These might, for example, be texts for unit and identifier. Depending on the data type of the I/O, the system verifies the syntax of each parameter value and that the numerical range is not exceeded.

Assigning global I/O parameters (block/nested chart)

How to edit the properties of all the I/Os of an object:

1. Double-click the object
   or
   select the object, followed by the menu command **Edit > Object Properties...**.
   The "Properties - Block" or "Properties - Chart" dialog box opens.
2. Open the "I/Os" tab.
   This tab contains a list of all the inputs and outputs.
   You can find additional information on this in the section ""Properties - Block/Chart" dialog box, "I/Os" tab (Page 437)".
3. You can sort the I/Os in the table before you set the parameters. This sorting order is temporary and restored to the original order when you close the dialog box.
   Click a column title to sort all I/Os again. Each click toggles the alphabetical order: ascending/descending. The sorting order of columns with selection boxes is determined by: not active / active. The "#" column indicates changes to the sorting order.
4. Make your entries in the fields that are not grayed out.

Configuring single I/Os

You change the properties of an individual I/O as follows:

Double-click the required I/O

or

Select the required I/O, followed by the menu command **Edit > Object Properties...**.

The "Properties - Input/Output" dialog box opens.

The dialog elements such as input fields and check boxes are displayed differently in the dialog. The display depends on the data type and the other properties of the I/O. Some of the elements may be read-only (for example "Block:") and some boxes may have the following properties:

- Not visible at all
- Visible but disabled (grayed out and cannot be edited)
- Enabled (can be edited)

Input of units

In the corresponding properties dialogs, you can either enter the unit input directly or select it via a drop-down list. The drop-down list contains all units that were installed as the basic set with CFC. The user can add to or modify these units according to the PCS 7 application.

You can find additional information on this in the section "Configuring Shared Declarations (Page 163)".

Advantages

Using the drop-down list for selection has the following advantages:

- The units benefit from consistent notation in the various applications (for example, process tag list, CFC Editor and SFC Editor).
- There is no need for the painstaking keyboard input of special characters such as "°C".

Color representation of a block or chart I/O

If a block or chart I/O has been changed, for example, due to configuration or interconnection, this status is indicated by a colored representation of this I/O.

The following display options are available:

- I/Os with values/attributes that have been configured by the user:
  This status is indicated by the font color of the configured value/text in the block icon.
- I/Os that have been assigned the default value, which have not been changed by the user:
  This status is indicated by the font color of the configured value/text in the block icon.
- For I/Os that were changed due to interconnection or configuration since they were last downloaded to the CPU:
  Status is indicated by the font color of the I/O name in the block icon.

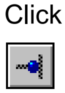
You can find additional information on this in the section "Blocks (Page 70)".

10.6.9.2 How to invert inputs

Interconnected binary inputs (of block and chart I/Os) can be inverted, In this way "0" becomes "1" and "1" becomes "0".

Inverting interconnected binary inputs

1. Select the input.
2. Click the button on the toolbar

   or
   select the **Edit > Invert Input....** menu command.

or alternatively

1. Double-click the input.
   The "Properties - Input/Output" dialog box opens.
2. Select the “Inverted” check box.

The input will be inverted and identified by a dot.

Canceling the inversion of an input

Execute the "Invert Input" function once again to cancel the inversion.

You can also cancel the inversion by performing the following actions:

- Delete the interconnection of the inverted input
- Move the interconnection to a different input
- Interconnect the input again

Points to note

Please observe the following information:

- If you copy a signal from an inverted input to another input, the new input will not be inverted automatically. The inversion must be done explicitly.
- If you move an interconnection from an inverted input to non-inverted input, the new input is not automatically inverted. The inversion is canceled at the input that is no longer interconnected.
- Inputs assigned to a chart I/O cannot be inverted.
  Remedy: Insert a NOT block.

10.6.9.3 Value identifiers at block/chart I/Os

Creating value identifiers

Using the value identifiers, you can define symbolic names (texts) for the parameter values of the block or chart I/Os. The system attributes below are available for the definition when creating block types or chart I/Os:

- "S7_enum"
- "S7_string_0" to "S7_string_25"

The system attributes below can be used for different values:

- "S7_string_0" and "S7_string_1" exclusively for binary values (data type: BOOL)
- "S7_string_2" to "S7_string_25" for integer values (data type: BYTE, INT, DINT, WORD, DWORD)
- "S7_enum" for binary values and integer values (data type: BOOL, BYTE, INT, DINT, WORD, DWORD)

Note

When using the system attribute "S7_enum", if needed, system attributes of the type "S7_string_0" to "S7_string_25" has no effect.

The system attribute "S7_enum" requires the object name of an enumeration as a value. The enumerations are included in the project as "shared declarations". Every enumeration can include several values. You can find information on this in the section: Configuring enumerations (Page 164)

The text for "S7_enum" and "S7_string_0/1" can have a maximum of 16 characters and for "S7_string_2...25" it can have a maximum of 8 characters. Only 8 characters are displayed. If you have more than 8 characters of text, you can define the characters to be displayed. You do this by using the "=" character in the text.

- If the equal sign (=) is included in the text, the first 8 characters to the right of the equal sign are displayed.
  Example: Motor=ON; Motor=OFF4567890; ON or OFF45678 is displayed.
- If the text does not include the equal sign, the first (left) 8 characters are displayed.

Displaying and modifying value identifiers in the chart

In the chart, you can specify whether to display the value identifier or the absolute value. Select the menu command **Options > Customize > Layout...**. In the “Customize Display” dialog box, you can set or reset the “Parameter: Value Identifier” option.

In the object properties dialog, you can select value identifiers defined for an I/O. The “Value” box will then contain an additional button for opening the drop-down list. From now on, the selected value identifier will be displayed at the I/O.

10.7 SFC in the CFC

10.7.1 Inserting SFC types in CFC charts

Analogous to block types, SFC types can be inserted into CFC charts from the block catalog. This creates SFC instances, which you can configure and interconnect. They are displayed as blocks, the same as CFC instances.

SFC types are listed in the catalog by family and in the "All blocks" folder. If the "Family" attribute is not assigned to the SFC type, it will be displayed under "Other blocks".

10.7.2 SFC external views in CFC charts

The SFC external view as of V6.0 replaces the control block SFC_CTRL.

In order to enable control of the SFC chart by means of CFC interconnections, it is assigned an interface derived from the interface of the SFC runtime system. It is displayed in the CFC as a block by a chart symbol, analogous to a nested chart. The block name is identical to that of the SFC chart and cannot be modified here.

The external view is displayed in a separate window of the CFC. In this window you cannot insert any additional objects, for example, blocks. Interconnections can be executed as usual in CFC, for example, textual interconnections, interconnections to shared addresses and block/chart I/Os.

When printing the external view locally, it is printed like a CFC chart without the interface.

Note

You can print interface I/Os in the form of a table in the SFC editor (print: external view).

10.8 Creating and working with interconnections

10.8.1 How to create block interconnections

General information about block interconnections

A block interconnection is the connection between the output of a block and one or more inputs of another or the same block. The data types of the input and output must be compatible.

Interconnected blocks can be located in the same sheet, in different sheets of the same chart or in different charts. The interconnection partner can also be located in another chart folder of the project or in another project of the same multiproject. You can find additional information on this in the section "Setting up AS-wide interconnections (Page 141)".

You can interconnect I/Os in the chart overview or sheet view.

A special form of a block interconnection is the textual interconnection, in other words, the interconnection partner can be located in a different chart folder.

A sheet bar entry is always generated when there is an interconnection to an object that is not located within the active sheet. If entries cannot be made in the sheet due to lack of space in the sheet bar, overflow pages will be created automatically.

A sheet bar entry is also generated the moment an interconnection within the sheet leads to a hidden I/O. The entry is labeled with the text "INVISIBLE" after the I/O name. This additional text in the sheet bar is available even when an I/O for interconnections across sheets is hidden.

Textual interconnections and interconnections to shared addresses or to runtime groups have no sheet bar entries when the I/O is hidden. These interconnections can only be recognized in the object properties of the block ("Interconnection" column of the "I/Os" tab). The same applies when both interconnection partners have been switched to invisible.

Note

Please observe the following information:

- If the output of an FB is interconnected to an INOUT I/O of an FC, the FC writes to the instance data block of the FB.
  When the FB reads this output in its algorithm (and not only writes), this may lead to problems or malfunctions of the output due to the FC.
- Note that the block outputs have specific default values and may affect other blocks when the CPU starts up if they have already been executed.

The correct **startup response** of the blocks is the responsibility of the project engineer.

Interconnections of an SFC instance

The "block contacts" characteristic of an SFC type affects the creation of an interconnection at the instance of the SFC type. Under certain conditions, all other block contact interconnections between the SFC instance and the block are automatically completed when an interconnection is created.

You can find more information on this in the section "Block icons" in the SFC online help or in the *SFC for SIMATIC S7* documentation.

Procedure

Create block interconnections as follows:

Single or multiple interconnections

1. Select the output you want to interconnect.
2. Click the input you want to interconnect to the selected output.
You can repeat this procedure to create multiple interconnections for this output.
You can select the I/Os in any order. You can also click the input first and then the output. Multiple interconnections also function in this way.

Sequential interconnections

1. Select the output you want to interconnect.
2. Press and hold down the <Shift> key and then click the target inputs you want to interconnect to the selected output, working in successive order.
With sequential interconnections, you must keep to the order output → input.

Interconnecting via drag-and-drop

1. Click the output you want to interconnect and hold down the left mouse button.
2. Drag the cursor to the target input.
You can select the I/Os in any order here. You can also drag an input to an output to interconnect them.

The interconnection is usually displayed as a line, in exceptional cases as a connector. This happens when a line cannot be drawn due to lack of space or if no additional entries can be made in the sheet bar.

Tips

- You can easily create interconnections across sheet or chart boundaries by opening the sheet view of several windows simultaneously.
- You can also create interconnections across sheet boundaries in the overview.

Additional information

You can find information about the display of interconnections in the section "Layout of interconnections (Page 76)".

You can find information about interconnections with S7 in the following sections:

How to interconnect to shared addresses (Page 143)

Working with textual interconnections (Page 149)

How to create and delete interconnections with runtime groups (Page 154)

Creating interconnections to chart I/Os (Page 154)

Interconnection rules for chart I/Os (Page 155)

Interconnection rules for data types A, DT, S, and ST (Page 160)

Interconnecting with SFC Charts (Page 148)

10.8.2 Setting up AS-wide interconnections

Introduction

With the CFC, you can set up interconnections to partners located on other automation systems. Interconnection partners are always block I/Os. Requirements for AS-wide interconnections:

- The PLCs involved are located in a common project or multiproject.
- The network configuration has been performed with NetPro.

Setting up AS-wide interconnections

You make the interconnection as you would for a chart-wide interconnection. To do this, open both charts and arrange them in the CFC in such a way that you can connect the source to the destination; in other words, click on the I/O to be connected in one of the charts and then click on the I/O partner in the other chart.

For connections, an interconnection line is drawn to the sheet bar. The AS-wide interconnection is marked as a green triangle in the small field of the sheet bar. The project/station/CPU type or hierarchical path and chart name/block/connection is entered in the large box.

Notes on AS-wide interconnections

- You can also perform a jump from the sheet bar to track signals with an AS-wide interconnection. The interconnection is then displayed flashing in the other chart.
- Only the S7 target system is supported.
- AS-wide interconnections to addresses and runtime groups are not possible.
- S7-300 CPUs are not supported.
- F programs are not supported and F blocks cannot be interconnected AS-wide.
- The fastest possible cycle for an AS-wide interconnection is 200 ms. In the case of an AS-wide interconnection, the cyclic interrupt OB30 - OB38 must be parameterized in such a way that its cycle times and priorities relate to each other. The faster the cycle time, the higher the priority must be.
- The maximum transmission volume between 2 partner CPUs in an OB is 4800 bytes.
- STRING and POINTER data types are not supported.
- An AS can communicate with a maximum of 15 partner AS.
- I/Os of nested charts cannot be interconnected AS-wide.
- AS-wide interconnections between an H station and an S7-400 station are not possible.
- Block contacts of all data types can be used for AS-wide interconnection only if the contact is configured as "IN" interface and cannot be used if the contact is configured as "IN_OUT" interface.

Maximum number of AS-wide interconnections

A maximum of 200 sending and 200 receiving AS-wide interconnections per OB and partner CPU are possible.

Note

It may no longer be possible in the current version to compile a program with AS-wide interconnections that could be run with the previous version. This is the case if the default of 30% maximum CPU load for AS-wide interconnections in this program has been exceeded. If you increase this default to the correct value, then you will be able to compile the program once again.

You can view and change the default in the dialog box "Settings for Compilation/Download".

Effects of moving a project

If you move a project out of a multiproject that is the source or destination of an AS-wide interconnection, the interconnection partner is recorded in the ES database. The interconnection to the sheet bar is retained; the sheet bar entry is deleted and replaced by a question mark.

When the project is reintegrated, the previously existing AS-wide interconnection is automatically reestablished upon update (F5) if the interconnection partner is available unchanged as before.

Note

Moving projects from the current (multi) project can lead to single-sided AS-wide interconnections if the AS-wide interconnection is deleted from a project but remains intact in the other when disconnected. If the last AS-wide interconnection is deleted from a project when disconnected, the inherent partner connection is deleted as well. As soon as the moved project is reconnected, a single-sided interconnection is created (identified by question marks in the sheet bar of the CFC). This single-sided interconnection then generates a warning in the code generator and is not loaded into the AS.

You can find information on eliminating one-sided AS-wide interconnections in: Synchronize AS-wide Interconnections (Page 432)

General information on the procedure

Once AS-wide interconnections have been created, the affected S7 programs need to be compiled and loaded. The S7 connection in NetPro must not be deleted manually. The NetPro IDs cannot be changed.

To view the NetPro IDs, follow these steps:

- Select the chart reference data. In the CFC click the button:

- In the "View" menu, select the menu command "AS-wide interconnection".

You can now read the NetPro IDs from the table.

The user does not need to undertake any special measures to perform the data transfer via the AS-wide interconnection. The blocks required for the data transfer are made available and loaded by the ES when loading is performed. These blocks are not instanced in the chart and cannot be seen in the catalog. The data transfer is called directly from the corresponding OBs of the AS (OB1, OB3x). The handling instructions are located in special DBs created by the code generator and transferred from the loader to the CPU.

The "IK STATE" block is located in the "ELEM_400" folder in the CFC library. This can be integrated into the program in order to respond to communication errors in the process. You can find a description of the "IK STATE" block in the Help for CFC elementary building blocks > AS-wide blocks > IK STATE

If the last AS-wide interconnection is deleted, the S7 connection in NetPro is also deleted. Following this, compiling and loading is required.

10.8.3 How to interconnect to shared addresses

Introduction

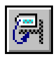
Shared addresses are interconnection partners outside the CFC charts, for example, shared data blocks, I/O signals, bit memories, timers, and counters. You can specify the addresses in symbolic or absolute form. You must, however, keep to certain conventions.

Interconnections to shared addresses are inserted and identified in the sheet bar by a small blue triangle.

Note

If the interconnection begins at an I/O that has been subsequently switched to hidden, the sheet bar entry is also deleted. You can then only recognize the interconnection in the object properties of the block/chart in the "Interconnection" column of the "I/Os" tab.

Procedure

1. Select the input/output and
   select the **Interconnection to Address...**
   command from the shortcut menu,
   or click the button in the toolbar
   

   The address selection list appears at the I/O, consisting of:
   - The input field (upper row) for editing the shared address
   - Table of all symbols in the current symbol table for the selected data type

   ---
   Note

   The address selection list will possibly be displayed in minimized format at the block I/O when you open it, in other words, only the input field is visible. Use the cursor to optimize the selection list size in order to display the entries in the symbol table also.

   ---
2. Specify the address as follows:
   - Enter the symbolic name or absolute address in the input field

   or
   - Select the symbol required.
3. Select a symbol or double-click the line of the selected symbol.
4. Press <Return> in the input field
   or
   click an area outside of the box.
   The address selection list is closed.
   Press <Esc> to exit without saving your selection.

Result

A sheet bar entry will be generated that refers to the target of the interconnection.

Note on input of the address

- Based on syntax rules (see Help on the symbol table, section "Permissible addresses and data types in the symbol table"), the CFC can detect whether you have input an absolute or a symbolic address.
- Absolute addresses (depends on the address type) can only be interconnected to block I/Os of the type BOOL, BYTE, WORD, or DWORD. Alternative data types (BYTE -> CHAR; WORD -> INT, S5TIME; DWORD -> DINT, REAL, TOD, TIME) can only be interconnected to symbolic addresses. For this purpose, the data type required is also specified in the symbol table (“Data type” column).
  See also "Absolute addressing without entry in the symbol table".

- The entry of an absolute address is not case-sensitive and may or may not include empty strings, for example, e1.1, E1.1, e 1.1. The permitted range of values depends on the type of CPU. Syntactical write errors in absolute addresses may cause the CFC to misinterpret the entry as a symbol!

Note

Always comply with CPU-specific limits. Otherwise you cannot download the compiled program to the CPU and a download error message appears.

- Symbolic addresses are enclosed by upper quotation marks in the sheet bar of the chart. You can thus check how the CFC has interpreted your entry, in other words, as a symbolic or an absolute address.
- The following addresses cannot be accessed via the CFC:
  OB, SDB, SFB, SFC, UDT and VAT
- Absolute access to DB elements is possible. The following samples show the permitted syntax:

Address	Access to
DB10.DW20	Data block 10, word 20
DB20.DX2.1	Data block 20, bit 2.1 The notation DB20.DBX2.1 is also tolerated as input, but "B" is not accepted and not shown in the sheet bar.

- Mixed addressing of DB elements is also allowed.
  Example: DB10.DW11, DB10.Valve, Mixer.DW8, Mixer.Valve
- ANY I/Os, for example, SD_1 for SFB 12/BSEND, can also be interconnected with a complete data block (shared address, absolute, for example -> DB1 or symbolic). The complete length of the DB is transferred. Entire data blocks can thus be transferred.
  The individual elements of the data blocks (even STRUCT) can be accessed through the interconnection to the DB element. This allows for the easy assembly and evaluation of the data to be transferred.
- If you want to use a symbolic address although the symbol has the syntax of an absolute address, you must enclose the symbol in double quotes.
  Example: "Q1.1", "DB10".Valve.

Accessing the symbol table

Call the menu command **Options > Symbol Table** to open the symbol editor for editing entries in the symbol table.

Please observe the STEP 7 conventions when you make entries in the symbol table.

Chart update

Select the menu command **View > Update** or press <F5> to update the names/addresses of operands and block types in the CFC charts of the active chart folder.

Important points on interconnecting

When interconnecting shared addresses, note the following relationships and options:

- You can interconnect block I/Os with an address in absolute or symbolic form. If the symbol is already defined in the symbol table and has a comment, you will see the symbol comment in the sheet bar.
  You can also enter the symbol in the symbol table at a later point in time. CFC recognizes the symbol when the chart is updated.
- If a shared symbolic address is not yet available in the symbol table, an error warning is output during compilation and the following actions will be triggered:
  - Input: The code is generated and the default value of the block type set for the interconnection.
  - Output: The interconnection is ignored in the generated code.
- If you want to change the context of the symbol, delete it from the symbol table and create a new one.
- Do not interconnect the EN input of startup blocks, for example, CONT_S, that are located in the process image, for example, I5.1, otherwise the startup function will not be enabled.
- If a STRUCT I/O of an instance block is interconnected with a shared data block and the structure name of the shared data block is subsequently changed, this change will not be detected during compilation. The change is updated in the ES data management only after closing and reopening the CFC chart.
  This also applies if the names of elementary data types are changed.
- If the name of symbolic and absolute address are identical, the absolute address is used during interconnection.

Modifying the interconnection to the address

Double-click the large field of the sheet bar (for this interconnection) to open the address selection list showing the default in the input field. You can modify (overwrite) this entry, delete it via the <Del> key or select another symbol from the list.

Absolute addressing without entry in the symbol table

Absolute addressing is also possible if a symbol does not exist in the symbol table for the absolute address or DB number.

Note the following rules:

- The address must not occupy the range that is reserved for CFC chart compilation.
- The address must not be located in a value range that does not exist for the current CPU.
- The address must match the data type of the linked element.

You can find more information on interconnections to shared addresses in:

Possible interconnections between symbols and data types (Page 147)

Examples of symbolic addressing (Page 147)

Examples of absolute addressing (Page 147)

10.8.4 Examples of symbolic addressing

Symbolic addresses

The following examples illustrate the options when specifying symbolic addresses:

Address	Remark
Limit switch	For example access to an input bit. The absolute address is defined in the symbol table.
Recipe. Setpoint	Access to a data block (recipe). The structure or the type of the data block determines the data block element (setpoint).
"I5.1"	Access to the symbol with the name E5.1. To avoid the absolute address with the same name being addressed instead of the symbol, the symbol is enclosed in quotation marks.

10.8.5 Examples of absolute addressing

Absolute addresses

The following examples illustrate absolute addressing:

Address	Remark
I5.1	Access to input 5.1
i5.1	Access to input 5.1
M6.7	Access to memory bit 6.7
MW10	Access to memory word 10
DB10.DW20	Access to data block 10, data word 20
DB20.DX2.1	Access to data block 20, bit 2.1 (syntax also: DB20.DBX2.1)

10.8.6 Possible interconnections between symbols and data types

Valid data types

The table shows which symbols with the specified data length can be interconnected to the block I/Os of which data types. You must then also enter the corresponding data type of the block I/O in the symbol table.

Data length	Address Symbol	Block I/O data type
8 bits	IB, QB, MB, PIB, PQB	BYTE, CHAR
16 bits	IW, QW, MW, PIW, PQW	WORD, INT, S5TIME
32 bits	ID, QD, MD, PID, PQD	DWORD, DINT, REAL, TOD, TIME

10.8.7 Interconnecting with SFC Charts

SFC access

A special form of interconnection is direct SFC access from actions or transitions to I/Os of the block or nested chart in the CFC chart. This type of SFC access can be "rewired" in the CFC chart, in other words, it can be moved from one I/O to a different I/O of a compatible data type. If the I/O is interconnected, you can decide whether only the SFC access or also the interconnection will be moved.

- **Move only the SFC access**
  Drag the selected I/O to the new I/O while holding down <Alt>. The SFC access symbol is moved to the new I/O. Existing interconnections will not be included.
- **Move interconnection and SFC access**
  Drag the selected I/O to the required new I/O. You will be prompted to include the SFC accesses. If you click “Yes”, the interconnection and the SFC access are moved. If you click “No”, only the interconnection is moved.
- **Delete the SFC access**
  You cannot delete an SFC access in the CFC.

Access markers

In the CFC chart, a marker at the I/O identifies SFC accesses. Write and read access are indicated differently.

Markers have the following meanings:

- A marker above the I/O = "read access"
- A marker below the I/O = "write access"

The color of the marker is identical to the interconnection color indicating a specific data type.

SFC references

You can display SFC references as follows: Double-click the block I/O to open the “Properties - Input/Output” dialog box. When I/Os with SFC access are used, this dialog box contains the “SFC Access…” button. Click this button to open an additional dialog box that contains a list of SFC references.

The reference contains the names of the SFC chart, SFC object (step name/transition name) and the object type (step/transition) and type of access (write/read).

Double-click the chart name of a listed object to open the relevant SFC chart where the referenced object is highlighted.

10.8.8 Working with textual interconnections

What is a textual interconnection?

A textual interconnection can only exist at one block/chart input and always refers to a block or chart output in the CFC. The textual interconnection remains "open" until it is converted into a "real" interconnection by closing it.

A textual interconnection you can close equates to the addressing of an input using a character string that identifies an interconnection source (output) through the pathname.

How is a textual interconnection created?

A textual interconnection is created as follows:

- Automatically as a path reference (chart\block.I/O) including the PH path (if it exists), by copying or moving it to a different chart folder.
  After copying, the textual interconnection is generated at the input of the copy; after moving, it is generated at the moved object as well as at the previous object, if the corresponding output is no longer located in the same chart folder. The interconnection is deleted at the output if the input is no longer available in the same chart folder.
- Via user action at the input: Select the input or open the shortcut menu and select the menu command **Insert > Textual Interconnection**.
  - As a path reference (chart\block.I/O) with PH path (if it exists).
    The interconnection is closed as soon as the interconnection partner is available in the chart folder. If the interconnection partner is not available in the chart folder until a later point, the interconnection can be closed via menu command **Options > Make Textual Interconnection**. A real block interconnection is created.
  - As an interconnection request (any character string that does not contain a defined path reference, for example, a comment).
    This interconnection request cannot be closed (warning in the log); the user must close it manually in the project.
- If the data type of interconnected I/Os no longer matches due to global modification of the type.

Note that a textual interconnection is created under the following conditions:

- When the source of an interconnection is deleted.
  The real interconnection is deleted in this case.
- When the interconnection source of internal interconnections to chart I/Os is deleted or moved (nested chart or block).
  The chart I/O assignment is lost in this case.
- When the global modification of the type involves renaming with interface modification.
  Note: The interconnection is retained if you only change names without modifying the interface.

Representation in the sheet bar

With an open textual interconnection, the interconnection partner does not exist in the chart folder or the reference does not correspond with an existing block I/O.
The large box of the sheet bar displays the one line of text you have entered.
The small box displays a yellow triangular identifier.

Note

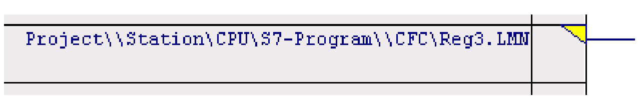
Please observe the following information:

- If the I/O with the textual interconnection has been hidden, there is no sheet bar entry. These interconnections can only be recognized in the object properties of the block ("Interconnection" column of the "I/Os" tab).
- The block with a hidden interconnection is indicated by a colored triangle in the top right corner of the block header.

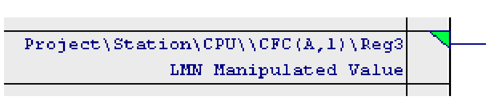
By closing the textual interconnection, you convert it to a real interconnection; that is, to a normal block interconnection. The small box no longer displays the identifier (yellow triangle).

Representation in the sheet bar for AS-wide interconnections

You can see the path to the AS-wide interconnection in the large field of the sheet bar. A yellow triangle is shown in the small field to indicate that the textual interconnection is open.



This is converted into a real interconnection by closing the textual interconnection. The yellow triangle in the small field is replaced by a green triangle.



With AS-wide interconnections, care must be given to the syntax of the path in the large field. It is important to distinguish whether or not the chart is integrated in the plant hierarchy.

- The syntax without plant hierarchy
  - Project\\Station\CPU\Program\\Chart\Block.IO
- The syntax with plant hierarchy:
  - Project\\Hierarchy Path\\Chart\Block.IO

Changing a textual interconnection

You can modify an existing textual interconnection as follows:

- Rewire it to another input of the same data type.
  You can find more information on this topic in the section How to modify (rewire) interconnections (Page 157).
- Edit the text in the sheet bar
  Double-click the entry in the sheet bar (small or large box) to open the dialog box in which you can enter your new text. The previous text is selected and can be overwritten.
- Replace the textual interconnection:
  - Via an interconnection to a shared address
    Select the input or text from the sheet bar then select the shortcut menu command **Interconnection to Address...**.
    You can find information on this in the section: How to interconnect to shared addresses (Page 143)
  - Via an interconnection to a chart I/O
    Select the input or text from the sheet bar and then select the shortcut menu command **Interconnection to Chart I/O...**. This dialog box shows a list that contains all configured chart I/Os of the active chart and the I/O type (IN, OUT, IN_OUT).

Closing open textual interconnections

Once all interconnection partners are available in the chart folder, you can convert the textual interconnections into real interconnections.

Use the menu command **Options > Make Textual Interconnections** to close all textual interconnections of the current chart folder.
You cannot close single textual interconnections.

The log generated lists all closed textual interconnections and any that are still open. Call this log with the menu command **Logs > "Textual Interconnections" tab** .

The following applies to compiling:

If one or several textual interconnections are not closed, a corresponding log entry is generated. You can view this entry with the menu command **Logs > "Compile" tab**.
The data is compiled and a substitute value is generated for the corresponding input (default value of the block type). An interconnection at the output is ignored.

Deleting and finding textual interconnections

Use the menu command **Options > Delete Textual Interconnections...** to open a dialog box that contains a list of all textual interconnections.

This dialog box offers you the following options:

- You can obtain an overview of all existing textual interconnections in the current chart folder. Click "Cancel" to exit the dialog box without deleting the interconnections.
- You can click "Go To" to open the chart in which the interconnection was implemented. The interconnection is displayed flashing.
- You can select textual interconnections and click "Delete" to delete them from the chart folder. The interconnections are deleted immediately without any prompt for confirmation.

General information

Please observe the following information:

- A textual interconnection consists of a maximum string length of 512 visible characters.
- A textual interconnection cannot co-exist with a closed interconnection or another textual interconnection.
- When you close a textual interconnection, all actions and error messages are listed and displayed in the log file. You can open the log file with the menu command "**Options > Logs > "Make Textual Interconnections" tab**.
- Open textual interconnections are tolerated when you compile the program. A warning is generated and the following actions are performed:
  - Input: The code is generated and the default value of the block type is applied to the open interconnection.
  - Output: The interconnection is ignored in the generated code.
- If an I/O with a textual interconnection is switched to invisible, the sheet bar entry is deleted. You can then only recognize the interconnection in the object properties of the block/chart in the "Interconnection" column of the "I/Os" tab.
- If an I/O is negated with a textual interconnection, this negation is not displayed in the technological editor.

Textual interconnection in the process object view

- Textual interconnections can be created in the component view, but not in the process object view.
- A textual interconnection that was created in the component view is interpreted as a global variable in the process object view. Because this variable does not exist, it is not recognized in the process object view and there is an error message "Interconnection to address not allowed". To close the textual interconnection in the process object view, the path reference (chart\block.I/O) needs to be entered.

Additional information

You can find more information on textual interconnections in the section:

Textual interconnections in branched and merged project data (Page 153)

10.8.9 Textual interconnections in branched and merged project data

Introduction

When branching projects into separate configurable segments and merging them again at a later point, note the following peculiarities and aspects that apply to the textual interconnections.

Sequence and procedure

When you split a project (which we refer to here as a "master project"), you copy parts of a chart folder to one or several other chart folders, for example, to temporary projects ("branch projects") for different project engineers. This action usually concerns the interproject transfer of one or several (but not all) charts. Interconnections across chart boundaries are converted into textual interconnections at the input side and deleted at the output side.

When you return the edited charts to the master project (copy or move), the application recognizes charts with identical names in the master project. In this case, you are prompted to overwrite the existing object. If you click "No", the chart is inserted as a copy ("None", if multiple charts are involved). If you click "Yes", the existing chart is overwritten ("All", if multiple charts are involved). The corresponding chart is deleted from the master project prior to the transfer of the chart from the branch project. In this case, contrary to a normal deleting action (<Del>), textual interconnections will be generated.
Select the menu command **Options > Make Textual Interconnections** to reconvert them into real interconnections.

Rather than copying charts to other projects, you can also move them using "Cut" and "Paste". Copying, nonetheless, has the advantage that the edited charts have the safety level of a fully functional (master) project until they are returned.

Note

Please observe the following information:

- You must not rename any of the interconnection partners in a "broken" interconnection, since it will be impossible in this case to close the textual interconnection.
- Modifications of charts in the master project are canceled if charts with identical names are returned from the branch projects to the master project.
- An interconnection might be generated unintentionally in the master project, for example, when you modify interconnections that exceed chart limits in the branch project and subsequently return only a single chart to the master project.
  Example: Chart CFC_A contains an interconnection to a block in the CFC_B chart. Both charts are copied to a branch project and edited further. In this case, the interconnection between both charts will be deleted. Only CFC_A is returned to the master project. A textual interconnection that you can actually close will be generated in the CFC_B chart of the master project. Result: The interconnection that was deleted in the branch project exists again in the master project.
- Textual interconnections generated prior to a copy/move action are applied to the target project. This can be a defined path reference (that can be closed) or any character string (interconnection request that is actually configured in the target project).

10.8.10 How to create and delete interconnections with runtime groups

General

You can enable and disable runtime groups dynamically. This means that the output value of a block decides whether or not a particular runtime group will be executed. In order to achieve this, connect the binary block output to the enable attribute of the runtime group. In this case, the setting of the “Active” option in the object properties of the runtime group is not taken into account.

A small red triangle in the sheet bar indicates interconnections to runtime groups.

Note

Please observe the following information:

- If the interconnection begins at an I/O that has been subsequently hidden, there is no sheet bar entry. You can then only recognize the interconnection in the object properties of the block/chart in the "Interconnection" column of the "I/Os" tab.
- The block with a hidden interconnection is indicated by a colored triangle in the top right corner of the block header.

Creating an interconnection

1. Select the binary output you want to interconnect, followed by the menu command **Insert > Interconnection to Runtime Group....**
   The dialog box "Insert Interconnection to Runtime Group" appears. This contains a list of all CPU tasks and runtime groups.
2. Select the task containing the required runtime group from the list.
   The list of runtime groups is sorted accordingly.
3. Select the runtime group and click "OK".
   The dialog box closes. The interconnection is entered in the sheet bar.

Delete interconnection

Click the small or large box in the sheet bar and press <Del>.

The interconnection to the runtime group is deleted. The interconnection is deleted automatically when you delete the runtime group.

10.8.11 Creating interconnections to chart I/Os

Interconnecting chart I/Os

You can create interconnections to chart I/Os from blocks or from nested charts with chart I/Os.

You have the following options:

- Interconnect I/Os to existing chart I/Os
- Create the chart I/Os at the same time as the interconnection

A small white triangle in the sheet bar indicates interconnections to chart I/Os.

You can find more information about chart I/Os in:

- How to create a chart with chart I/Os (Page 98)
- Interconnection rules for chart I/Os (Page 155)

10.8.12 Interconnection rules for chart I/Os

Interconnection rules

The following rules apply when interconnecting chart I/Os:

- Block I/Os interconnected internally cannot be interconnected with a chart I/O.
- A block input of the type IN can be interconnected with chart I/Os of the type IN or IN_OUT.
- You can interconnect a block I/O of the data type ANY with chart I/Os of any elementary data type (exception: not with the data type ANY and POINTER).
- The compatibility of data types of the I/Os you are interconnecting may differ on the various target systems. If there is a data type incompatibility, an error message is displayed when you make the interconnection.
- You cannot configure chart output parameters.
  Remedy: Set parameters for the output in the block. This value is then entered in the corresponding chart output.
- Structured data types of FC outputs cannot be interconnected with chart I/Os.

10.8.13 Signal tracking

Tracing interconnections

Even if numerous interconnections are displayed in a sheet, you can nevertheless track the path of a specific interconnection as follows:

- Click the displayed line or the large field in the sheet bar.
  All lines and sheet bar entries, including those in overflow pages, connected to the selected line will flash on-screen. Signal tracking is also possible along branched lines and beyond break connectors.
- From the chart I/O of a nested chart, you can track the signal to the I/O that is interconnected internally with this chart I/O.
  In the shortcut menu, select the command **Track Signal**. This opens the nested chart; the interconnection and the sheet bar entry flash.
  You can track an interconnection that leads into a nested chart over the sheet bar to the interface by opening the shortcut menu for the I/O or the sheet bar entry and selecting the menu command **Jump from Sheet Bar**. The higher-level chart is then opened and the relevant I/O of the nested chart is selected if it is not interconnected. If the chart I/O is interconnected, the interconnection flashes.
- To disable the flashing, simply click a different position in the chart.

Signal tracking is also possible with Jump from Sheet Bar.

You can find additional information on this in the following section: Jump from Sheet Bar (Page 156)

10.8.14 Jump from Sheet Bar

With “Jump from Sheet Bar”, you can move quickly to the interconnected block on the other sheet of the current chart or another chart from a sheet bar entry of a sheet or an overflow page:

- If you jump from the sheet bar entry or from an I/O with a single interconnection, the jump is made directly.
  The sheet or chart in which the interconnected block or nested chart is located opens. With block interconnections and interconnected chart I/Os, signal tracking is activated (line flashes). If the I/O of a nested chart is not interconnected, the relevant chart I/O is selected.
  If the destination of the jump is an overlapping block, the block alone is fetched to the center of the screen and selected. Since I/Os and connecting lines cannot be displayed, signal tracking is not possible in this situation.
- If you jump from the sheet bar from an output with a multiple interconnection, a dialog box opens with a list of the interconnections of this output. The jump is executed when you double-click the output required or select it and confirm with “OK”.
- You can use the **Edit > Go To > Jump Back** menu command to return to the point at which you started the jump even if you have closed the selected chart or have deleted the block.

You can find additional information in the section: Signal tracking (Page 156)

10.8.15 How to copy interconnections

Procedure

In order to copy an interconnection, proceed as follows:

1. Select the interconnected input.
2. Press and hold down the <Ctrl> key and the left mouse button and drag the cursor to the target input (of the same type). When you reach this input, release the mouse button first and then the <Ctrl> key. An additional interconnection is thus created.

Copying is a highly convenient function, particularly in situations when the interconnection originates in a block from a different chart, since you do not need to initially locate the source.

Note

An SFC access cannot be copied in CFC.

10.8.16 How to modify (rewire) interconnections

Procedure

You can modify existing interconnections in the sheet view. To do this, move the interconnection from one I/O to another I/O with a compatible data type. Proceed as follows:

Input

1. Position the cursor on the input whose interconnection you want to move to another input and hold down the mouse button.
2. Drag the cursor to the target input (of the same type) and release the button. The interconnection line is redrawn.

Output

The procedure is basically the same as for the inputs. However, the following rules apply:

- If there is more than one interconnection, all interconnections will be moved to this output.
- If the interconnection from an output to an address is dragged to an output that is already interconnected to the same address, a warning is displayed. You can then decide whether or not to delete the interconnection of the original block.

  Note

  Please observe the following information:

  - If you want to modify an interconnection that is accessed by an SFC chart (see the block I/O identifier), a message prompts you to move the SFC access also. If you click “Yes”, the interconnection and SFC access are moved, if you click “No”, only the interconnection is moved.
  - If you only want to move the SFC access to a different I/O of a compatible data type, drag the I/O with the SFC access to the required I/O, holding down the ALT key.

10.8.17 How to delete interconnections

Procedure

Proceed as follows to delete an interconnection:

Multiple interconnection

Deleting all interconnections of an output:

1. Select the output or the connection line and select the menu command **Edit > Delete** or press <Del>.
2. If you click "Yes" in response to the prompt for confirmation, the interconnections are deleted.
   If you click "No" in response to the prompt for confirmation, no interconnection is deleted.

Deleting one of several interconnections of an output:

1. Select the input and press <Del>
   or
   position the cursor on the sheet bar entry of the output and press the right mouse button.
   The small field before the sheet bar entry is selected.
2. In the shortcut menu, select the command **Delete Interconnection(s)**.
   The interconnection to the input will be deleted.
   You can also click the small field in the sheet bar directly and delete the relevant interconnection via <Del> or the menu command **Edit > Delete**.

Single interconnection

You can delete the interconnection for the output, input, sheet bar entry, or connecting line as follows:

Select the corresponding object and
select the menu command **Edit > Delete**
or
press <Del>.

Textual interconnection

Single textual interconnection:

You delete a single textual interconnection from the chart in the same way as a single interconnection (see above).

All textual interconnections:

1. Select the menu command **Options > Delete Textual Interconnections**.
   A dialog box opens which contains a table listing all textual interconnections of all charts in your current chart folder.
2. In the "Chart" column, select all the textual interconnections you want to delete.
3. Click "Delete".

Note

An SFC access cannot be copied in CFC.

10.8.18 Data types and structures

10.8.18.1 Data types for S7

The following topics list all the S7 data types that can occur at a block/chart I/O that is capable of being assigned parameters or being interconnected.

The description shows the following elements:

- Abbreviation of the data type
- Keyword and data type
- Value range
- Sample entries
- Display with the maximum length in the chart (examples)

The maximum string length for displaying parameters in the chart is limited to 18 characters. This ensures that sufficient space is maintained for blocks and interconnections. Input options are not affected by this limitation.

Data types

- ANY, A (Page 425)
- ARRAY (Page 425) (not permitted in CFC)
- BOOL, BO (Page 426)
- BYTE, BY (Page 426)
- CHAR, C (Page 427)
- COUNTER, CR (Page 427)
- DATE, D (Page 427)
- BLOCK_DB, DB (Page 425)
- DINT, DI (Page 428)
- DATE AND TIME, DT (Page 427)
- DWORD, DW (Page 428)
- BLOCK_FB, FB (Page 426)
- BLOCK_FC, FC (Page 426)
- INT, I (Page 428)
- POINTER, P (Page 428)
- REAL, R (Page 429)
- STRING, S (Page 430)
- STRING[N], SN (Page 430)
- STRUCT, ST (Page 430)
- TIME OF DAY, T (Page 431)
- S5TIME, T5 (Page 429)

- TIME, TI (Page 430)
- TIMER, TR (Page 430)
- WORD, W (Page 431)

10.8.18.2 Interconnection rules for data types A, DT, S, and ST

Interconnection rules

The following interconnection rules apply to block I/Os of the structured data types ANY, STRING, DATE_AND_TIME, and STRUCT:

	Data type	
Interconnection type	ANY	STRING, DATE_AND_TIME, STRUCT
Mandatory interconnection for:	FC - OUT, with all other I/Os, NIL is transferred, which must not be evaluated by the calling block (only relevant for STL blocks written by the user; with PCS 7 blocks, this is guaranteed).	FC - OUT, FC - IN_OUT, FB - IN_OUT (with STRUCT)
Target/source of the intercon-nection (**address**):	Any: CFC block I/O, Address ranges, DB elements, Entire DBs	DB elements, CFC block I/O (except for FC - OUT)
Target/source of the intercon-nection (**data type**):	Any: except for same data type (ANY)	only to same data type

Interconnection to chart I/O

Structured I/Os of FBs and structured inputs of FCs can be freely interconnected to chart I/Os. Since structured data types of FCs do not have their own resources (storage space in the instance DB), FC outputs cannot be interconnected with chart I/Os.

Note

Blocks of STEP 7 or PCS 7 libraries do not contain FCs with structured outputs.

10.8.18.3 Structures

Layout

Structures are used to group data in a tree view and apply those data to a block I/O. Structures in the block view are assigned, similar to elementary data types, an I/O name and the type "ST".

A structure consists of several elements and a nesting depth of up to 8 levels. The following elements are permitted:

- Elementary data type (BO, WORD, etc.)
- Structure

A structure element contains the following information:

- Type (of the elementary data type, otherwise "ST")
- Name
- Value (only with the elementary data type)

Interconnecting

You can only interconnect a block I/O with a structure to other structures, not to elementary data types. The structures you interconnect must be compatible, in other words, the order, data type, and the element name of the elementary data types it contains must be identical. The names of the structures can be different.

You can only interconnect the complete block I/O structure, not separate elements of the structure.

Note on interconnection of a structure (STRUCT)

- To simplify the combination of APL blocks and standard blocks, the following structures represent an exception because they only consist of two elements:
  - Name.VALUE [BOOL]
    Name.ST [INT]
  - Name.VALUE [REAL]
    Name.ST [INT]

  "Name.VALUE" can be interconnected in these structures with the elementary data types BOOL or REAL.
  "Name.ST" (interconnection: STRUCT> elementary data type) is not evaluated or amended to "Name.ST:= 16#80" (interconnection: elementary data type->STRUCT).
- Global data block, AS-wide interconnections, F-parameters and CFC I/Os are not supported.

Interconnections to shared addresses

You cannot interconnect a block I/O with a structure to a shared address that refers to an entire data block (DBx). Interconnections to structures within the first level of the DB are possible (DBx.name_st). Interconnections of elementary data type I/Os are permitted with elementary data types within the first level of the DB (DBx.element).

Note

If a STRUCT I/O of an instance block is interconnected with a shared data block and the structure name of the shared data block is subsequently changed, this change will not be detected during compilation. This also applies if the name of the elementary data type is changed. The change is updated in the ES data management only after closing and reopening the CFC.

Editing structure elements

You can display the object properties of the structure or of an elementary data type in the structure and edit them (assign parameters).
Double-click the block I/O to display the "Select Structure Element" dialog box. In the structure that appears, select the element you want to edit and open the properties dialog by clicking the "Properties" button or by double-clicking the element.

You can open the dialog box in both the edit and test mode.

Note

You cannot interconnect FC outputs with a structure to chart I/Os.

Inserting structure in dynamic display

You can insert the entire structure or individual structure elements in the dynamic display. To do this, proceed as follows:

1. Open the dynamic display with the menu command **View > Dynamic Display**.
2. Select the block I/O in the CFC and select the "Insert in Dynamic Display" command in the shortcut menu.
   The "Select Structure Element" dialog box opens.
3. Select the top level if you want to insert all structure elements or select an individual element of you only want to insert a single element.
4. Click "Apply".
   The dialog box closes and the selection is inserted in the dynamic display.

10.9 Shared Declarations

10.9.1 Configuring Shared Declarations

Introduction

In the project, you can store shared declarations that can be used by different applications. These shared declarations are, for example, enumerations, units and equipment properties. This ensures that attributes with the same names are used throughout a project.

Configuring

When a new project is created, the shared declarations and the “Enumerations”, “Units” and “Equipment properties” folders they contain are created automatically. Shared declarations are also created in the master data library when a new multiproject is created.

In the component view or plant view of the SIMATIC Manager, you create the "Shared declarations" folder directly below the project node. With the shortcut menu command **Insert New Object > Shared Declarations**, you insert the folder with the underlying folders "Enumerations", "Units" and “Equipment properties” into the project.

You can delete, copy, move, and create the folders contained in the “Shared declarations” folder individually. This folder must not contain more than one folder of the same type.

When you select the “Shared declarations” folder, the following menu commands are available via the shortcut menu command “Insert New Object”:

- Enumerations
- Enumeration
- Units
- Unit
- Equipment properties
- Equipment property

You can also insert the "Enumeration", "Unit" or "Equipment property" objects if no “Enumerations”, “Units” or “Equipment properties” folder exists. It is then created automatically at the same time.

If you insert one of these folders, an existing folder of this type is overwritten if you confirm the displayed prompt. It is also overwritten if you renamed the existing folder earlier.

After insertion, the “Enumerations” folder already contains the enumeration, with the name “Operating State” but without values. The enumeration is used in PCS 7 for hiding messages.

The "Units" folder already contains a list of units installed with CFC. You can extend or modify these as required.

No equipment properties exist yet in the "Equipment properties" folder. In the multiproject/ master data library, this is the folder where you insert a type. In the plant hierarchy, it is where you insert an instance of the equipment property.

Additional information

You can find additional information about this in the following sections:

Configuring enumerations (Page 164)

Configuring units (Page 166)

Configuring equipment properties (Page 166)

Updating shared declarations in the multiproject (Page 167)

10.9.2 Configuring enumerations

Introduction

You use enumerations for block I/Os or chart I/Os that have the system attribute "S7_enum". Regardless of the current value of the system attribute (value matches the object name of an enumeration), you can use any object name from the enumerations that exist within the project. These enumerations are available in a drop-down list in the object properties of the I/O.

Enumerations are permitted for I/Os for the following data types with Boolean and integer values: BOOL, BYTE, INT, DINT, WORD, DWORD.

Note

If a system attribute "S7_string_0" to "S7_string_25" **and** the system attribute "S7_enum" is assigned to the I/O, only the enumeration will be used as the value identifier (system attribute "S7_enum").

Creating an enumeration

When a new project is created, the shared declarations and the “Enumerations” folder they contain are created automatically with an enumeration. Shared declarations are also created in the master data library when a new multiproject is created.

The enumeration with the name “Operating State” is intended to automatically hide messages but does not contain any values.

An enumeration can be generated in the following ways:

- By creating it in COMOS Integrated Engineering and subsequently importing it into PCS 7.
  When imported into PCS 7, enumerations are stored in the folder "Shared Declarations > Enumerations" of the master data library. The enumerations in the project are not automatically synchronized in this case. Instead they are included in the comparison in the data transfer dialog and can also be adapted by this.
  Alternatively, the enumerations can be synchronized in master data library and project by the user. You can find more information on synchronization in the section "Updating shared declarations in the multiproject (Page 167)".
- Creation in the master data library or project.
  This setting is described below.

Principle of creation

You can create a new enumeration in the master data library or project by selecting the “Shared declarations” folder and selecting the shortcut menu command **Insert New Object > Enumeration**. Alternatively, you can use the menu command **Insert > Shared Declarations > Enumeration**.

You can change the object name and the display name in the object properties of the enumeration.

You can specify the object name of the enumeration so that it matches the value of the system attribute "S7_enum" of the block I/O for which this enumeration will be used.

Note

In the object properties of a block I/O or in the process object view, the "Enumeration" box is active when the system attribute "S7_enum" is assigned to the I/O. It does not matter whether the value of the system attribute matches the existing name of the enumerations in the project. You can assign the required enumeration to the I/O at any time using a drop-down list.

The display name of the enumeration can be translated into different languages and is transferred to WinCC when you compile the OS.

When you create the enumeration, "0.1" is entered automatically as the version. You can change the version in the object properties of the enumeration.

You can assign any number of values to the enumeration. For enumerations that will be used for I/Os with the data type BOOL, only two values are practical.

Inserting values

You create a new value using the shortcut menu command **Insert New Object > Value** or the menu command **Insert > Shared Declarations > Value**.

The first value to be entered contains the numeric value 0. This number is incremented with each additional value added.

The display name of the value is displayed in the object properties of the block I/Os under "Value". You can select each existing value of an enumeration there from a drop-down list.

If you want to identify the enumeration in SIMATIC BATCH for a control strategy, select the “Control strategy” option in the "Properties Enumeration” dialog box, “General” tab. When you select the option, the symbol of the enumeration and the value changes its appearance.

Checking plausibility

You can check any enumerations in the multiproject/project to ensure that the enumeration values are unique, in other words, that they have not been assigned more than once.

You can find additional information on this in the section "Checking the plausibility of shared declarations (Page 168)".

10.9.3 Configuring units

Inserting a unit

Create a new unit by selecting the "Units" folder and the shortcut menu command **Insert New Object > Unit**. Alternatively, you can use the menu command **Insert > Shared Declarations > Unit**.

If you want to undo a change to the units or remove added units, insert the "Units" folder again. The content of the existing folder is then overwritten with the basic set from CFC.

The object name is the text that is displayed as the unit. In the object properties, you can change the following:

- Name
- Version
- Author
- Comment

10.9.4 Configuring equipment properties

Creating an equipment property

When a new project is created, the shared declarations are created automatically along with the associated equipment properties folder. You use the equipment properties (unit parameters) in BATCH projects.

Create a new equipment property by selecting the "Equipment properties" folder and the shortcut menu command **Insert New Object > Equipment Property**. Alternatively, you can use the menu command **Insert > Shared Declarations > Equipment Property**.

Take the following into account when creating equipment properties:

- The equipment property that is created in the shared declarations is an equipment property **type**.
- The equipment property that is created in the plant hierarchy is an **instance** of the equipment property.

You can generate as many equipment property instances as required from each equipment property type.

Configuring the equipment property type

You configure the equipment property type in the shared declarations.

1. In the "Equipment properties" folder, select the equipment property, followed by the shortcut menu command "Object Properties".
2. Assign the equipment property a logical name as well as a display name.

3. Select the required data type from the drop-down list.
   Depending on which data type is selected, the "Enumeration" and "Unit" input fields will either be enabled or disabled.
   If the "LOCATION" data type is selected, three additional check boxes will be displayed: Source, Dest and Via.
   Note: If all three check boxes are disabled, this equipment property is ignored by SIMATIC BATCH.
4. Make any other settings you may require in the light of how you intend to proceed in SIMATIC BATCH.
5. Click "OK" to complete configuration.

Configuring an equipment property instance

You configure equipment property instances in the plant hierarchy.

1. Select the "Unit" hierarchy folder followed by the shortcut menu command **Insert New Object > Equipment Property**.
2. Select the "Equipment properties" folder, followed by the shortcut menu command **Insert New Object > Equipment Property**.
3. Select the equipment property, followed by the shortcut menu command "Object Properties".
4. In the "Name:" box, select the equipment property type from the drop-down list.
   The instance are now automatically assigned the same name as the type from Shared declarations.
5. You can edit the author and comment in accordance with your requirements.
6. If the equipment property contains an enumeration, you can use the "Value" drop-down list for the instance to assign a configured value to this enumeration.
7. Click "OK" to complete configuration.

10.9.5 Updating shared declarations in the multiproject

Updating shared declarations

You have two options for performing the update:

- **Merge the shared declarations of all projects in the multiproject**
  The enumerations, units and equipment properties are updated across all the available projects. This means that at the end of this process, all the shared declaration objects are present in each of the projects concerned.
- **Export shared declarations of a project to another project**
  The shared declarations of a template project are exported to selected destination projects of the multiproject.

Procedure

1. Select the multiproject, followed by the shortcut menu command **Shared Declarations > Update in Multiproject....** Alternatively, you can select the menu command Options > Shared Declarations > Update in Multiproject... .
   A dialog box opens where you can select the type of update.
2. Select either a) "Merge the shared declarations of all projects in the multiproject" or b) "Export shared declarations of a project to another project".
   a) Merge the shared declarations of all projects in the multiproject
   – Click "OK" to start the update.

   b) Export shared declarations of a project to another project
   – Select the project (or library) in the list that you want to use as a template for exporting the shared declarations to another project.
   – Click "OK".
   Another dialog box opens. It lists all the projects that are available for selection as the destination project.
   – If you do not want the shared declarations of the template project to be transferred to a specific project, you must remember to deselect the project concerned.
   – Click "OK" to start the export.

Note

Please observe the following information:

- A master data library may only be used as the source of an update.
- Changing names for units, enumerations or equipment properties in the template causes these to be created new in the target project when the comparison is made. The previously used shared declarations are retained at the points of use.
- Units, enumerations or equipment properties deleted in the template are ignored during the comparison in the target project and are therefore retained there.

Result

Once the update is complete, the result is displayed in the form of a message. In the event of an error, a full log will be generated.

You can view the full log by selecting the shortcut menu command **Shared Declarations > Display Full Log....**

10.9.6 Checking the plausibility of shared declarations

Plausibility check

You can check any enumerations in the projects and libraries of a multiproject to ensure that the enumeration values are unique, in other words, that they have not been assigned more than once.

You perform this plausibility check for each individual project or a library. When an update in the multiproject is performed, consistency is checked across the entire multiproject as a matter of course.

If you select a multiproject, all the projects and libraries within this multiproject will be checked one after the other.

To initiate the check, proceed as follows:

1. Select a multiproject, a project or the shared declarations.
2. Select the **Shared Declarations > Validate** command from the shortcut menu.

Results of the check

- If you have run the check on a multiproject, a message will be displayed to indicate that the process is complete. This message will ask you to open the local project logs so that you can see if any errors have occurred.
- If you have run the check on a single project, the log will be displayed automatically once the checking process is complete. If any errors have occurred, you will find them recorded in the log, e.g., "Enumeration type 'xyz': 'value 1 assigned several times".

10.10 Configurations for OS and MIS/MES

10.10.1 Operator control and monitoring

Introduction

Messages are generated on the CPU during process runtime and need to be passed on to the WinCC operator control and monitoring system.

Via the message configuration in CFC you can define event-relevant messages, edit their texts and assign attributes directly in the block.

In your AS configuration you have already specified data required on the OS for OS-AS communication, in other words, for operator control and monitoring. You transfer this data to the OS during OS compilation.

Message blocks

After you have inserted a message block into the CFC, a message will be generated automatically. This block has a predefined message structure with default attributes and message texts. In this way, the CPU sends the appropriate message from the AS when an event occurs without additional configuration on the part of the user. The signals that form messages can also be assigned associated values, which allow the input of dynamic values in message texts.

You can edit those attributes, such as message class, message type, and message texts for each block instance via the message configuration functions (Special properties: "Messages"). If the "Messages" button is disabled, the block does not have message capability.

Operator control blocks

You can set and edit WinCC attributes for blocks with inputs that allow operator control. To do this, open the dialog box by clicking "Operator Control and Monitoring..." in the object properties of the block.
You can modify this property at the block instances intended for operator control and monitoring by selecting or clearing the "OCM possible" option in the "Object properties" section of the block.

Templates for block icons

In a WinCC template picture, you can create multiple block icons for each block type as a template. The templates of the block icons and their variants are located in the picture "@PCS7TypicalsBasisLibraryV8.pdl".

In this way you can indicate specific variants of a type, for example, the MOTOR block as motor, fan or pump.

In the "Block icon" input field of the "Object properties" dialog, you can define the block icon to be displayed for this block instance in WinCC. You can enter the variant of the block icon as a variant number or variant name. A maximum of 16 characters is possible.

The variants of a block icon can be distinguished by the "/" character at the end of the icon name, for example, <icon name>/2 ".

Variant	Remark
1	Standard style PCS 7
2	Style similar to APL style. Also read the note below!
<Variant name>	User-defined variants; user-defined variant name as the name of a block icon; max. 16 characters

The input field is active if the "Create block icon" check box is selected. The input field is disabled if you clear the "Create block icon" check box. However, the entry is retained.

Note

Templates of block icons and their versions

The templates of the basic block icons and their variants are located in the "@PCS7TypicalsBasisLibraryV8.pdl" picture. As described in the table, variant "2" in this picture means that the style of variant "2" is similar to the style of the APL!

The icons in this picture should not be confused with the icons for the APL, since the block icons and variants of the APL are found in the "@PCS7TypicalsAPLV8.pdl" picture.

AS <-> OS communication

After you have configured the messages, the OS data required for AS <-> OS communication will be downloaded to one or several target operator stations where they are used by graphic objects or faceplates.
The "AS-OS Engineering" software package must be installed to enable this transfer. Start the compilation in the SIMATIC Manager using the menu command Options > "Compile Multiple OSs" wizard > Start... or, for a single OS, using the menu command Edit > Compile.

Note

Note than when you assign names to charts/blocks, the string length of the variable name must not exceed 128 characters for OS compilation. This name is made up of the following components:

- The name of the folder in the hierarchy path
- The chart name
- The block name
- The separator (period)
- The I/O name

10.10.2 Configuring archive tags

Introduction

In CFC, you can select the block I/Os intended for operator control and monitoring (system attribute S7_m_c := 'true') for archiving in WinCC.

The I/Os identified as being relevant for archiving are created as archive tags during compilation of the OS in the OS project. If it does not already exist, a process value archive is also created automatically.
Requirement: In the "Compile (Multiple) OSs" wizard, the option "Archive tags" is selected and in the object properties of the OS, the setting "Create/update archive tags" is selected.

If an I/O relevant for archiving is deleted later or is changed to "No Archiving", the corresponding archive tag is deleted the next time you compile the OS.

If there are no more archive-relevant I/Os available, the process value archive is deleted, if it also contains no more archive tags.

Configuration

You identify tags for archiving in the object properties of the I/O (for a single I/O) or on the "I/Os" tab in the object properties of the block (for multiple I/Os).

In the drop-down list of the input field, you choose whether and how the I/O will be archived:

- "No Archiving" (S7_archive := 'false')
  The I/O will not be archived (any longer).
- "Archiving" (S7_archive := 'shortterm')
  The I/O will be archived on the OS or on an archive server.
- "Long-term archiving" (S7_archive := 'longterm')
  The values archived on the OS or on the archive server will be stored for long-term archiving on CD, DVD, MOD, tape, etc.

The system attribute S7_archive is permitted for all OCM-capable I/Os of the data types BOOL, BYTE, WORD, DWORD, INT, DINT, and REAL.

Note

If a new block type is used in a project, the existing block instances remain unchanged in terms of the system attribute "S7_archive". New instances of the block type will inherit this attribute.

10.10.3 Configurations for MIS/MES

Introduction

The enterprise levels MIS and MES require specific information from the automation level. To ensure that only information relevant for MIS/MES is transferred, the projects can be created with a suitable filter.

The system attribute "S7_mes := "true"" is used as the filter. It is set for blocks and block I/Os intended for operator control and monitoring (S7_m_c := "true"). The system attribute is specified by the block type and can be modified in the block instances.

Configuration

You have the following configuration options:

- In the "Blocks" dialog box, "General" tab, you can activate or deactivate the entire block for the MIS/MES system by selecting or deselecting the "MES-relevant" check box. The option can only be changed when "OCM possible" is set.
- You modify the system attribute for I/Os that can be operator controlled and monitored as follows:
  - For several I/Os: In the "Properties - Block" dialog box, in the "I/Os" tab
  - For a single I/O: In the "Properties - Input/Output" dialog box

10.11 Defining runtime properties

10.11.1 Runtime properties

Note

When you create a new chart, a runtime group is created automatically in which all the blocks of this chart are integrated.

Runtime properties of blocks

The runtime properties of a block determine how the block is executed in the run sequence within the entire structure of the CPU. These properties are decisive for the response of the target system in terms of reaction times, dead times, or the stability of time-dependent structures, for example, closed loops.

When it is inserted, each block is assigned default runtime properties. It is thus inserted into the run sequence of a task. In S7, a task consists of an OB (organization block).

You can find additional information on installation pointers in the section "Concept and use of installation pointers (Page 179)".

If required, blocks can also be installed in the runtime groups of tasks.

Note

When you create a new chart, a runtime group is created automatically in which all the blocks of this chart are integrated.

Runtime groups

Runtime groups are used for structuring tasks. The blocks are installed sequentially in the runtime groups and can be assigned "reduction ratio" and "phase offset" attributes in the "Object Properties" dialog box.

You can find additional information on attributes in the section "Runtime attributes (Page 197)".

Runtime groups can be enabled and disabled individually (for example, by a block output of the "BOOL" data type). The blocks of a disabled runtime group will not be executed.

You can find additional information on this in the section "How to create, edit and delete runtime groups (Page 186)".

Chart-based runtime group management

With the "chart-based runtime group management", the blocks of a CFC are automatically managed in separate runtime groups based on the chart, i.e. the blocks of the CFC are organized in runtime units that are permanently assigned to this CFC.

You can find additional information on this in the section "Chart-based runtime group management for blocks of CFCs (Page 192)".

Additional information

You can find more information about runtime properties in:

- Displaying runtime properties (Page 175)
- Modifying the run sequence and the installation pointer (Page 182)
- Concept and use of installation pointers (Page 179)

See also

Display of feedback in the signal processing (dead time) (Page 177)

10.11.2 Displaying runtime properties

Introduction

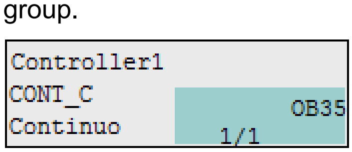
There are various ways in which you can display information on runtime properties:

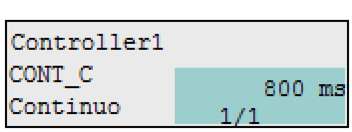
- Of a single block
- Of the entire CPU

Runtime properties of individual blocks

The runtime properties of each block are displayed on a colored background in the block header, in the runtime display field:

- Upper line (left):
  Identifier for the Display of block execution (Page 198)
  Task name (for example, OB 35)
- Upper line (right):
  Name of the task or sampling time of the block (depends on the display setting).
  The sampling time is the task cycle time multiplied by the reduction ratio of the runtime group.





- Lower row (to the left of the slash):
  position of the block or runtime group in the task
- Lower row (to the right of the slash):
  if the block is installed in a runtime group, position of the block in the runtime group; otherwise "-"

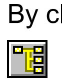
Tip: When you position the cursor on the box of the runtime properties, the tooltip box will display the task name and, if the block is installed in a runtime group, the name of the runtime group. The task name is also displayed if the sampling time is entered in the runtime properties field.

Entire CPU

You can obtain a complete view of the run sequence as follows:

1. Double-click the runtime properties box in the block header.
   The runtime editor is started and the task in which the block is installed (first insert position) opens in the run sequence window. The relevant block is highlighted.
2. Select the menu command **Edit > Run Sequence...**
   or
   Click the icon:
   

   The runtime editor opens with the window of the run sequence. If a block was selected, the task in which the block is installed (first insertion position) is opened in the run sequence. The relevant block is highlighted.
   In this window, you can also edit the run sequence.
3. Select the menu command **Options > Chart Reference Data...**
   or
   Click the icon:
   

   An application is started with its own window and menu commands/toolbar buttons.
   The "Run Sequence" view shows the complete run sequence of the current CPU with comments and runtime attributes.
   You have the following options for enabling the "Run Sequence" view:
   – With the menu command **View > Run Sequence**
   – By clicking the icon:
     

   The tasks containing the objects are open. You can print out a list of the run sequence.
   In this window, you cannot edit the run sequence.

10.11.3 Display of feedback in the signal processing (dead time)

Definition of terms:

Feedback in an interconnection occurs when the function/block with the interconnection target is arranged before the function/block of the interconnection source in the program sequence (run sequence).

Undesired feedback leads to downtimes in the signal flow or incorrect calculation results.

Overview

The sequence of the blocks is checked during manual insertion of an interconnection between blocks. When the "Avoid implicit dead time" function is enabled, the run sequence, when possible, is automatically corrected in order to resolve detected feedback in the interconnection and therefore avoid dead times in the signal processing.

- **Principle of correction:**
  Basically, a distinction is made between the following optimization functions:
  - Optimization of groups (Optimize groups/tasks...) including all blocks in the runtime editor.
    This optimization is started in the runtime editor in the context menu of a runtime group using the menu command "Optimize groups/tasks".
  - Parallel procedure during an interconnection process for avoiding feedback / dead times.
    For this parallel procedure for avoiding feedback / dead times, the "Avoid implicit dead times" must be activated in the CFC Editor.
    This function is described below.

- **Check:**
  With a manual interconnection process, a check is first carried out whether a feedback path / dead time occurs.
  A detected feedback path / dead time is graphically displayed with a symbol on the input side of the interconnected block at the respective interconnection.

  Note

  Even if the "Avoid implicit dead times" function is deactivated, the check is performed on the feedback path and a detected feedback path is displayed at the interconnection.

- **Automatic correction:**
  If a feedback path is detected and the "Avoid implicit dead times" function is activated, the blocks involved are then exchanged accordingly in their run sequence if they are in a shared runtime group.
  If the blocks involved are not all in a runtime group, a correction is then performed.
  Feedback detected is then implicitly resolved during the interconnection process.

- **No automatic correction**:
  Even if the "Avoid implicit dead times" function is activated, no correction of the run sequence is performed during the interconnection process, if no feedback or an irresolvable feedback is detected.
  If the feedback cannot be resolved, this is indicated by the symbol described below.

- **Switch off automatic correction:**
  Automatic correction with the "Avoid implicit dead times" function can be deactivated. Switching off of automatic correction then makes sense if no changes to the runtime properties are desired.
  Activation or deactivation of the function is described in the following section.

- **Objects involved:**
  The procedure for avoiding feedback paths / dead times takes the following objects into consideration:
  – The two blocks involved in the new interconnection.
  – All dependent, i.e. interconnected blocks which constitute an isolated group without external dependencies.
  – Blocks which were already moved in the run sequence can still be taken into consideration when feedback paths / dead times are eliminated, if they continue to constitute an isolated group of interconnected blocks without external relationships.
- **Objects that are not involved:**
  – F programs generally are not involved in the correction procedure.
  – Already existing interconnections, with or without feedback, generally remain unchanged in terms of their state, i.e. the relative run sequence of already interconnected blocks remains unchanged.

Note

Notes on the "Avoid implicit dead times" function when working in CFC

Before working in the CFC editor, check the setting of the "Avoid implicit dead times" function in the menu command or on the icon in the toolbar. If necessary, deactivate the function to avoid undesired automatic corrections.

Enabling/disabling the function "Avoid implicit dead time"

To activate or deactivate the function, select the menu command "Options > Avoid implicit dead times" in the CFC editor.

The function is toggled by selecting the menu command "Avoid implicit dead times".

Alternatively, the function can also be activated/deactivated via the symbol in the toolbar of the CFC Editor.

The current status is displayed on the left of the menu item or on the symbol of the toolbar.

Displaying feedback

Feedback is graphically displayed to the user:

- The following symbol is shown on the input side of the interconnected block at the respective interconnection in the CFC.

  This gives the user visual feedback during the interconnection process.

You can find additional information on this in the section "Layout of interconnections (Page 76)".

User response

To delete the status display for the feedback, you need to check and adjust the run sequence of the blocks in the runtime editor.

Note

The graphic display of feedback is a configuring tool for the user.

Feedback that occurs in the CFC may be desired/required. It may also not be resolvable due to the system, or only by removing the interconnection.

10.11.4 Concept and use of installation pointers

Concept of the runtime model

The runtime model as of V6.0 allows chart-oriented structuring of the run sequence. All blocks are inserted in successive order in the run sequence. The installation pointer determines the insertion position for the next runtime object to be inserted in the run sequence. We differentiate between chart installation pointers and block installation pointers.

The following exist in an S7 program:

- A global chart installation pointer for the program
- A separate local block installation pointer for each chart.

In conjunction with the runtime group created automatically for each chart, the concept of the installation pointer ensures better chart orientation. This feature, for example, allows block-oriented handling of charts blocks when branching and merging a project for multi-user engineering. This allows you to word chart-oriented and to specify the runtime sequence of the blocks regardless of the runtime sequence of other charts.

Provided the user does not specify otherwise in the installation pointers, charts are inserted as follows:

- New charts are inserted after existing charts.
- Blocks are inserted in the chart directly one after the other.

Other advantages:

- The insertion position cannot be adjusted unintentionally by project engineers during chart-oriented multiuser engineering.
- There is no need to modify the complete OB when a chart is modified. Rather, only the corresponding runtime group of the modified chart is compiled and loaded.
- The load performance of the CPU can be influenced according to the specific chart.
- Individual charts can be enabled/disabled during CPU runtime (also in test mode), without the need to initialize this action by means of an external control (for example, SFC).

Chart installation pointer

Each program contains its own global "chart installation pointer". This pointer determines the following:

- The task into which the runtime group of the next new chart created is inserted.
  When copying a chart, the chart insertion pointer remains unchanged and is not taken into account because the sequence properties of the source it was copied from are applied.
- The position after a runtime group or block at the task level at which the runtime group of the chart created next is inserted. If the task does not yet contain a runtime group, this position is "empty". The pointer refers to the actual task.

This pointer is customized only in the runtime editor (not in the CFC Editor). OB 35 is the default chart installation pointer.

You customize the chart installation pointer as follows:

1. Select a task, a block/SFC or a runtime group in the run sequence.
2. Select the menu command **Edit > Predecessor for Insertion Position**.

Display of the chart installation pointer

The predecessor for the insertion position is highlighted visually in the run sequence by means of a light-green background.

Block installation pointer

Each CFC (including nested charts) contains its own local "block installation pointer". This pointer determines the following:

- The task into which the next new block is to be inserted.
- The position after a block at which the next new block is placed in the runtime sequence.
  The block position always has a unique link to the task number.

The block installation pointer is "undefined" in an empty chart. Instead, the chart installation pointer will be displayed in the status bar. After insertion of the first block, this information is fetched from the current chart installation pointer.

The block installation pointer is customized only in the CFC Editor (not in the runtime editor). The default start position (after a new chart was created) is the beginning of the automatically generated chart runtime group.

You customize the block installation pointer as follows:

1. Select a block from the chart.
2. Select the menu command **Edit > Predecessor for Insertion Position**.

If you insert an additional block in the chart, the installation pointer is moved to this block. It becomes the new "Predecessor for Insertion Position".

Note

Behavior of the block installation pointer during copying and pasting

With standard functions, e.g. AND, OR, etc., the block installation pointer behaves as it does during copying to the CFC when it is inserted, for example from the template catalog.

When blocks are copied to the CFC, they are always copied to the end of the runtime group. The block installation pointer is not included in this process and remains unchanged.

Display of the block installation pointer:

The predecessor for the insertion position differs from other blocks in the chart because it is highlighted in color. The box for the runtime properties in the block header is displayed in a black font on a light-green background.

Note on nested charts:

Blocks copied/inserted in nested charts are always located at the end of the runtime group. The position of the block installation pointer is ignored.

Additional information

You can find information about migration in the section: Migration of old projects to the enhanced runtime model (Page 38)

10.11.5 Modifying the run sequence and the installation pointer

Starting the runtime editor

Call the menu command **Edit > Run Sequence...** and click on the following icon in the toolbar to open an additional window:

This window is split into a hierarchy window (left) and a detail window (right). The structure of this window is similar to that of Windows Explorer and is handled in the same way.

Moving objects

To move an object (chart, runtime group or block), select it in the window (right or left pane) and then drag it to the object after which you want to install it.

When you drag an object to a runtime group:

- The object is installed at the first position within the runtime group if the tree is expanded [-].
- The object is installed after the runtime group if the tree is closed [+].
- If the runtime group is empty, you will be asked whether or not to install the block within the runtime group. If you click "Yes", it is installed **inside**, if you click "No", it is installed **after** the runtime group.

Any object you drag to a block/chart within the runtime group is inserted after this block/chart.

When you drag an object to a task, it is inserted **before** the existing installed objects.

Note

When you move blocks, verify that – according to runtime model V6.0 and higher – all chart blocks exist only in their corresponding runtime group. If you move them to another group, the chart-oriented structure would no longer exist, making segmented editing of charts harder or impossible for multiuser engineering.

Removing a block

You can only delete blocks from a task if they are installed more than once in the run sequence. At least one insertion point must be retained. Removal is denied if only one block was inserted. Otherwise, the block is deleted and the run sequence of the next blocks is adapted accordingly.

Inserting blocks

You can install several instances of the blocks and runtime groups in different tasks using the copy / paste function. Use either the corresponding menu commands, or the toolbar buttons, or drag-and-drop while holding down the <Ctrl> key (for additional information, refer to the "Moving objects" section).

You can also install blocks (if you have the windows displayed simultaneously) by dragging these directly from a CFC to the required position in the run sequence.

Note

Please observe the following information:

- Installation of a block in different tasks implies that you should not install several instances of the block in a cyclic task. However, the system does not prevent this installation.
- The objects with system identifier "**@**" which were inserted automatically in the run sequence during generation of the module drivers may only be edited using the **Options > Charts > Generate Module Drivers...** function in the SIMATIC Manager; these objects may neither be moved nor deleted manually.

Customizing installation pointers

You can edit the installation pointers as follows:

- **Chart installation pointer** (default: OB 35)
  To edit the chart installation pointer, go to the runtime editor and select the corresponding task (OB), or a block at the task level (not within a runtime group), or a runtime group within the task.
  Select the **Edit > Predecessor for Insertion Position** menu command in the runtime editor.
  The predecessor for the insertion position is highlighted using a font with light-green background.
- **Block installation pointer**
  You cannot set up the block installation pointer in the runtime editor. To edit the block installation pointer, open the CFC Editor and select the block for the default insertion position for all successive blocks.
  Select the **Edit > Predecessor for Insertion Position** menu command in the chart.
  The insertion pointer is incremented with each new block you insert. The last block inserted will become the predecessor for the installation position. The insertion pointer is not modified when you copy blocks.
  If you delete the block you have defined as predecessor for the insertion position, the block insertion pointer is decremented, which means it is set to the position of the previously inserted block. This also applies if the block is moved to a different chart. The block insertion pointer of the destination chart is not modified. The moved block retains the insertion position it had in the previous chart.

You can find additional information on the run sequence in the section: How to find the insertion position (Page 185)

10.11.6 Optimizing the run sequence

Purpose of optimization

This function is used to optimize the program run sequence according to the data flow in order to reduce dead times to a minimum during program runtime in the CPU. Tasks and runtime groups are optimized separately.

Start

To start the optimization process, in the runtime editor, select the menu command

- **Options > Optimize Run Sequence** to optimize the entire run sequence.

Alternatively, if you only want to optimize the selected task or runtime group, open the shortcut menu with the menu command

- **Optimize Groups/Tasks**.

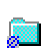
Selecting single elements

When optimizing the run sequence, you can optimize specific elements or exclude them from optimization. Select the setting for the selected task via its object properties. The "Properties - Task" dialog box appears.

- You can optimize the full task, including all enabled runtime groups.
  Option "Task and runtime groups" (default setting)
- You can optimize only the enabled runtime groups of a task.
  Option "Runtime groups only"
- You can exclude the full task - including its runtime groups - from optimization.
  Option "None"

Note

Please observe the following information:

- The selections made in the "Properties – Task" dialog box will not affect optimization in cases where this process is started via the shortcut menu command **Optimize Groups/Tasks** for an individual runtime group or task.
- You can enable specific runtime groups for optimization via the object properties dialog of the corresponding runtime group (option "Optimize run sequence").
  This means that you can exclude specific runtime groups from optimization by deselecting this option.
- All F objects are excluded from the optimization.
- Tasks or runtime groups excluded from the optimization are identified by an additional element in the symbol.
-  : If "None" or "Runtime groups only" is selected for optimization of the run sequence in the object properties of the task.
-  : If the "Optimize run sequence" check box is not selected in the object properties of the runtime group or optimization is not activated for the task.
- The contents of runtime groups generated by the driver generator ("Generate module driver") (@......) will not be optimized, since the correct order has already been set for these groups.

If the optimization is carried out after the generation of the module driver, there is no guarantee that the runtime groups of the driver blocks are still in the order set by the driver generator. Therefore, the module driver is started again during the next compilation procedure.

What happens during an optimization run?

All tasks are optimized separately. The runtime groups of a task are treated separately. The reduction ratio and phase offset of a runtime group are not taken into consideration.

The data flow volume is determined by the number of interconnections. This includes all block-to-block interconnections, as well as interconnections to SFC charts and between module outputs and the ENABLE signal of a runtime group. Global and textual interconnections will be ignored. Interconnections to the chart interface are taken into account up to the actual interconnection source. If no such connection exists, the interconnection will be ignored, in other words, if the interconnection terminates at an interface. Interconnections to blocks in other tasks, or accesses from SFC charts to block I/Os are also ignored.

Interconnections to or from a runtime group are treated as if they were interconnections of the runtime group itself. The runtime groups forms a virtual block at task level. Interconnections between internal blocks of a runtime group are only used for optimization processes within the runtime group. On the one hand, this ensures a correct arrangement of the actual runtime groups, and on the other hand an optimal position of the runtime group within the task.

Subsequent optimization runs are carried out without effecting unnecessary changes in order to reduce the volume of compiled and downloaded configuration data.

Note

When the block is connected via the INOUT parameter, the data flow may be reversed from the input to the output. This cannot be taken into account during optimization of the run sequence.
Remedy: In this case, you have to optimize the run sequence manually and remove the corresponding runtime group from the optimization.

10.11.7 How to find the insertion position

Procedure

In the runtime editor you can search for the insertion positions of an object as follows:

1. Select the menu command **Edit > Find...**.
   A dialog box opens.
2. Use the option buttons to select the search object (task, group, SFC, or block).
   You can enter the full name or only its initial letter(s).

Result

The search result is displayed in a window. If no object matching the search key was found, a message is displayed.

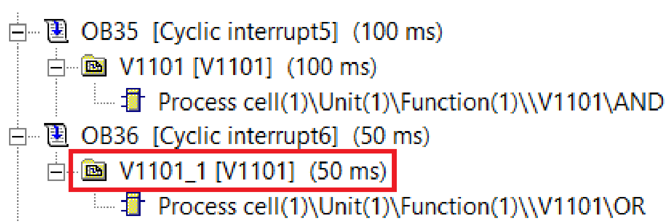
10.11.8 How to create, edit and delete runtime groups

What are runtime groups?

Runtime groups are optional but are created automatically when you create a new chart. Runtime groups are used for structuring or splitting tasks. The blocks are installed in sequential order in the runtime groups.

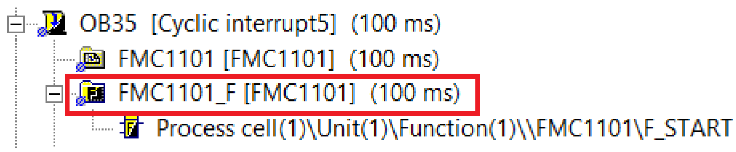
The following runtime groups are used in the run sequence:

- **Standard runtime groups:**
  There are two types of standard runtime groups:
  - Created manually in the run sequence.
    For example:
    E1101 (100 ms)
  - Created automatically when a chart is created.
    For example:
    Chart-based (managed runtime group): V1101 [V1101] (100 ms)
    Non-chart-based (unmanaged runtime group): M1101 (100 ms)
- **Extension runtime groups:**
  In addition to standard runtime groups, these runtime groups are created automatically when blocks from standard runtime groups (chart-based) are moved to other cyclic OBs.
  For example:
  "V1101" is the standard runtime group inserted in OB35 (by default). In the run sequence, if you move a block (for example, OR) manually from OB35 to OB36, a new extension group named "V1101_1" will be created in OB36 and the block will be inserted in this runtime group.

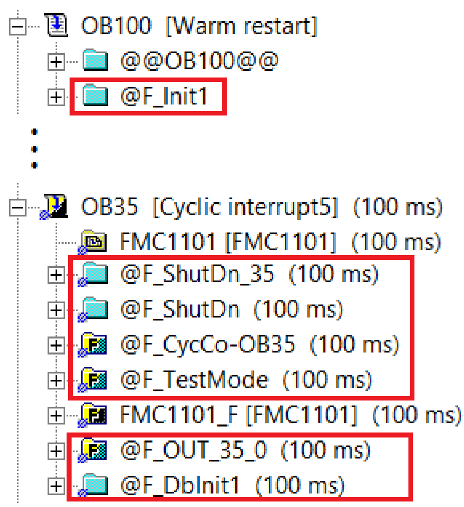


- **Runtime groups created by the driver generator:**
  Created automatically during generation of the module drivers when a program is compiled.
  For example:
  @PA_CPU
- **Runtime groups created by the system in OB 100:**
  Created automatically by the system when a program is compiled.
  For example:
  @@OB100@@

- **Fail-safe runtime groups:**
  In addition to the standard runtime groups, these runtime groups are created automatically in the same OB when fail-safe blocks are inserted in the charts. In the run sequence, fail-safe blocks are inserted in corresponding fail-safe runtime groups and other blocks are inserted in the standard runtime group.
  For example:
  “FMC1101” is the standard runtime group for the chart. If you insert a fail-safe block (for example, F_START) in the chart, a new fail-safe runtime group named “FMC1101_F” is created in the same OB and the block is inserted in this runtime group.



- **Fail-safe runtime groups created by the system:**
  In addition to the standard runtime groups and fail-safe runtime groups of fail-safe charts, these runtime groups are created automatically (depending on the fail-safe block) when fail-safe programs are compiled.
  For example:



You can use the runtime groups to:

- Delete or reconnect selected blocks for execution on the AS.
  Runtime groups are activated or deactivated using a block output of the data type "BOOL" or, with runtime groups that are not interconnected, using the "Active" option in the object properties of the runtime group.
- Execute selected blocks with a specific reduction ratio, in other words, every n-th number of cycles, and/or with a phase offset. This improves the load balance on the CPU.

- You can arrange tasks containing a large number of installed blocks in smaller units. In the improved runtime model V6.0 and higher this is achieved by automatic and chart-granular insertion of chart blocks in the runtime groups (one chart per runtime group).
  Advantage: Instead of creating one "large" FC when you compile each task, "smaller" FCs are created, depending on the number of runtime groups.
  If the program is modified later, only the runtime groups/FCs that actually contain modified blocks are given the "modified" ID. That is, the time required for compiling changes and downloading changes online will be reduced considerably.
- All blocks with startup relevance are installed in runtime groups in OB 100.
  When the blocks are installed, there is a search for existing runtime groups. If none are found or if the maximum number of blocks for this runtime group has been reached, a new one is created. The names of the runtime groups are created generically, for example "@@OB100@@(2)".

You can assign runtime attributes to objects you have inserted in the runtime groups. You can find additional information on this in the section "Runtime attributes (Page 197)".

Runtime groups in chart-based runtime group management

With the "chart-based runtime group management", the blocks of a CFC are automatically managed in separate runtime groups based on the chart, i.e. the blocks of the CFC are organized in runtime units that are permanently assigned to this CFC.

No block of another CFC can be included in the runtime groups of this CFC. This reduces the dependencies for selective downloading.

When instantiating blocks with startup characteristics during creation of a chart, a corresponding runtime group is automatically created in the startup task OB 100 in addition to the runtime group in the standard task (for example, OB 1) .

Recognition of a CFC in which chart-based runtime group management is activated:

- You can recognize the assignment of the runtime group to the CFC with chart-based runtime group management by the fact that the name of the associated CFC in the runtime group is in parentheses in the runtime editor.
- For CFCs with chart-based runtime group management, the "Chart-based insertion" option is selected in the properties of the chart folder or the CFC.

Integration and migration of CFCs:

- When the "Chart-based insertion" option is selected in the properties of the chart folder, chart-based runtime group management is automatically activated when a new CFC chart is created.
- For PCS 7 projects created as of version PCS 7 V8.1, this "Chart-based insertion" option is automatically enabled. For an existing PCS 7 project that was created with a version prior to V8.1, this option only works on newly created charts after the option has been enabled.
- Chart-based runtime group management is not automatically activated for existing CFCs. To activate chart-based runtime group management in existing CFCs, the menu command "Charts > Chart-based runtime group management" is available in the SIMATIC Manager from the shortcut menu of the chart folder or individual charts. The associated runtime groups are also adapted automatically.

You can find additional information on chart-based runtime group management and migration of existing CFCs in the section "Chart-based runtime group management for blocks of CFCs (Page 192) ".

How are runtime groups created?

Runtime groups can be created in two ways:

- Automatic
  Runtime groups of cyclic tasks (e.g. for CFC): After you have created a new CFC, a runtime group is automatically generated and inserted according to the chart installation pointer.
  The properties of automatically generated and manually created runtime groups differ:
  The automatically generated runtime group and the chart form a dependent relationship that is maintained until the name of the runtime group is modified.
  This dependent relationship ensures identical names both of the runtime group and of the chart. The name of the runtime group is automatically updated when the chart name is modified.
  When this dependent relationship no longer exists due to modification of the runtime group name, the automatically generated runtime group responds as any other manually inserted runtime group.
  OB 100:
  If blocks with startup characteristics are used, they are installed not only in the cyclic OB but also in OB100 in a runtime group created by CFC. A maximum of 10 blocks are installed in a runtime group. When more blocks need to be installed, additional runtime groups are created.
  Chart-based runtime group management:
  When chart-based runtime group management is activated in the CFC, the maximum number of 10 blocks per runtime group no longer applies.
  Driver generator:
  When creating module drivers, runtime groups are created with the identifier "@..." and these can then only be processed by the driver generator.
  Chart-based runtime group management:
  When the "Chart-based insertion" option is selected in the properties of the chart folder, creating a new CFC when instantiating blocks with startup characteristics automatically creates a corresponding runtime group in startup task OB 100 in addition to the runtime group in the standard task (OB 1).
  You can find additional information on this in the paragraph above, "Runtime groups in chart-based runtime group management".
- Manual
  You can create runtime groups manually at any time using the menu command **Create New** or **Copy**. The name of a runtime group you copy and paste into the run sequence is appended a consecutive number without brackets.

Creating a new runtime group

In order to create a runtime group, proceed as follows:

1. Select the menu command **Edit > Run Sequence...** or click the button

   The runtime editor opens and displays the run sequence window.
2. In the left window, select the task in which you want to install the runtime group or an object as the default insertion position for the runtime group.
3. Select the menu command **Insert > Runtime Group...**.
   The "Insert Runtime Group" dialog box opens.

4. Enter the name and comment in the corresponding boxes.
   You can enter the name in either upper or lower case letters.
5. Enter the values for the runtime attributes. If this function is not supported by the target system, the attributes "reduction ratio" and "phase offset" are deactivated and displayed in gray.
6. If necessary, modify the preset options:
   – Optimize run sequence (information available in the section: Optimizing the run sequence (Page 183))
   – Active (the runtime group is active for editing in test mode)
     Exception: "Active" is always set in F systems.
7. Close the dialog by clicking "OK".
   The runtime group is inserted at the required point.

Naming a runtime group

The maximum permitted length of the name is 22 characters.

It is recommended to use only 18 characters to provide some reserve for automatic renaming, for example, by the automatic addition of the character string "(1)" to avoid name collisions during copying.

Runtime group	Naming convention
Standard runtime groups (chart-based (managed) or non-chart-based (unmanaged))	Chart name is assigned as the runtime group name.
Extension runtime groups (extended runtime groups in other cyclic OBs)	Chart name suffixed with the extension "_x" (where x is 1, 2, 3, 4, and so on) is assigned as the runtime group name.
Fail-safe runtime groups	Chart name suffixed with the extension "_F" is assigned as the runtime group name.

Note

If the name already exists during copying:

If you copy a runtime group and paste it, the chart name suffixed with "(x)" (where x is 1, 2, 3, 4, and so on) is assigned as the name of the copied runtime group.

For example:

If there exists a runtime group named "ChartName" and if you create a copy of this runtime group, the copied runtime group is named as "ChartName(1)" because the name "ChartName" already exists. Further, if you paste it again (create another copy), the next copied runtime group is named as "ChartName(2)". But if you create a copy from the "ChartName(2)" runtime group then the copied runtime group will be named as "ChartName(2)(1)".

Note

If the maximum length of the name is exceeded during copying:

If you copy a runtime group which is named with 22 characters (maximum length) and paste it, the name of the copied runtime group is shortened from the end of the string until "~(x)" can be appended (where x is 1, 2, 3, 4, and so on).

For example:

If there exists a runtime group named "ChartNameMotorControlA" (22 characters) and if you create a copy of this runtime group then the copied runtime group is named as "ChartNameMotorCont~(1)".

By the general rules of naming in the case where the name already exists, the new name should have been "ChartNameMotorControlA(1)" (25 characters) but this exceeds the maximum permitted length of 22 characters. Therefore, the name is shortened to 22 characters with the methodology explained above.

Editing a runtime group

1. Select the runtime group in the left or right window and then select the menu command **Edit > Object Properties...**.
2. In the "Properties - Runtime Group" dialog box, modify the existing values (name, comment, phase offset, reduction ratio).

Do not edit runtime groups with the "@..." identifier. Those objects are created when the module drivers are generated and should only be manipulated by the driver generator. When they are no longer required, they will be deleted again by the driver generator.

You can edit the runtime groups created in OB100 (@@OB100@@...) by the system.

Note

Renaming a runtime group

- If you rename a runtime group, the chart name will not change.
- If you rename a chart, the new chart name is assigned as the name of the standard runtime group.
  Note: In the case of non-chart-based (unmanaged) runtime groups, this is applicable only until the standard runtime group is renamed. Once the standard runtime group of an unmanaged chart is renamed, renaming of chart will not have any effect on the runtime group name.

Activating and deactivating a runtime group

You can activate or deactivate runtime groups for editing by selecting/deselecting the "Active" check box in the run sequence. This function only affects runtime groups that are not interconnected.

Deleting a runtime group

You can only delete runtime groups that do not contain objects.

1. Select the runtime group and then
   select the menu command **Edit > Delete** or press <Del>.
2. With the menu command **Edit > Delete Empty Runtime Groups**, you can delete all empty runtime groups of the current S7 program after confirming a prompt.

Note

If the deleted runtime group was selected as the "Predecessor for Insertion Position", the default insertion position is moved to the next element of the task before the deleted group or, if no additional element exists, to the start of the task.

10.11.9 Chart-based runtime group management for blocks of CFCs

Overview

During configuration, the blocks of a CFC are automatically installed in a cyclic task for the "standard program". In addition, the blocks are installed in the corresponding runtime groups in accordance with the task list of the block type.

The CFCs and runtime groups are defined independently of one another, because the runtime groups in which the block calls are programmed have no clear reference to a CFC. This is true even if, when you create a CFC, a corresponding runtime group is implicitly created and an "installation pointer" to the runtime group or a block in this group is set.

It is also not guaranteed that exclusively the blocks of one CFC are included in an runtime group.

Chart-based runtime group management

With the "chart-based runtime group management", the blocks of a CFC are automatically managed in separate runtime groups based on the chart. This means the blocks of the CFC are organized in runtime units that are permanently and exclusively assigned to this CFC.

- This requires that "Chart-based insertion" is enabled in the properties of the chart folder or the CFC. You can find additional information on this in the paragraph below, "Integration of new CFCs ...".
- Assignment of the CFC to the runtime group:
  For a CFC with chart-based runtime group management, the runtime group for the "standard program" has the same name as this CFC.
  You can recognize the assignment of the runtime group to the CFC by the fact that the name of the associated CFC in the runtime group is in parentheses in the runtime editor.
  No block of another CFC can be included in the runtime groups of this CFC.
  The corresponding runtime groups are created depending on the blocks used in the CFC.

- Naming the runtime group of a CFC:
  The runtime group of the standard program in the cyclic task is given the name of the CFC. The runtime groups of acyclic tasks receive the name of the standard program, which is suffixed with the next available number, for example "(1)".
  When you rename a CFC with chart-based runtime group management, the associated runtime groups are renamed as well.
- Block with startup characteristics:
  When instantiating blocks with startup characteristics during creation of a chart, a corresponding runtime group is automatically created in the startup task OB 100. The runtime group of the CFC in the OB 100 automatically receives the next available name based on the name of the CFC.
- Effects:
  - Improved handling of download units for each CFC (SCD - selective download to the PLC) and the local execution optimization for each individual chart with a better overview.
  - Automatic management of standard and F-program, if F-blocks are used.
  - The programs of a CFC are thus managed separately from each other, although their blocks are displayed together in the data flow chart of the CFC.
- Coexistence;
  CFCs with and without activated chart-based runtime group management can coexist in a project.
- Requirement for the "Selective download" function:
  Chart-based runtime group management is optional, but it is required to use the "Selective download" function for CFCs/SFCs.
  Selective download is only possible for those charts in which chart-based runtime group management is activated. Inserting the new blocks of a CFC in the runtime group of this chart reduces the number of dependent charts which would also need to be included in the selective download.

Using F-blocks:

When F-blocks are also used in a CFC with chart-based runtime group management, not only is a runtime group for blocks automatically created in the "standard program", but also a runtime group for the included F-blocks, the "F-program".

- It is no longer necessary to manually move the F-blocks to a separate runtime group in this case.
- The name of the runtime group for the F-blocks contains the name of the CFC and the suffix "_F".

Display in the runtime editor

The display of runtime groups in the runtime editor depends on whether chart-based runtime group management is activated in the corresponding CFC.

	Chart-based runtime group management is not activated for the corresponding CFC.
	Chart-based runtime group management is activated for the corresponding CFC.
	Chart-based runtime group management is not activated for the corresponding CFC with F-blocks.
	Chart-based runtime group management is activated for the corresponding CFC with F-blocks.

Integrating, migrating and managing CFCs in the chart-based runtime group management

- Integration of new CFCs
  When the "Chart-based insertion" option is selected in the properties of the chart folder, chart-based runtime group management is automatically activated when a new CFC is created.
  For PCS 7 projects created as of version PCS 7 V8.1, this "Chart-based insertion" option is automatically enabled. For an existing PCS 7 project that was created with a version prior to V8.1, this option only works on newly created charts after the option has been enabled. When enabled in the properties of the chart folder, this option has an effect on all charts in the chart folder.
- Migration of existing CFCs
  Chart-based runtime group management is not automatically activated for existing CFCs. The menu command "Charts > Chart-based runtime group management" is available for this purpose in the SIMATIC Manager from the shortcut menu of the chart folder or individual charts. The associated runtime groups are also adapted automatically.
  The entire chart folder or several selected charts can be migrated.
  As of PCS V6.x, runtime groups have been created automatically in the OB 100 in addition to the cyclic runtime groups. A maximum of 10 blocks were placed in these runtime groups. To reduce the dependencies in a selective download, the runtime groups in OB 100 should be created or re-sorted based on the chart. The effort required for switching to the chart-based runtime group management is less than in previous versions. Nevertheless, beforehand you should determine if reducing dependencies for selective downloading justifies the effort required for resorting the existing runtime groups in OB 100.

Note

Runtime groups after the migration

During migration of CFCs to the chart-based runtime group management, the associated runtime groups are also adapted automatically. This step can also result in runtime groups that are empty and are not included in the chart-based runtime group management. These runtime groups are not automatically deleted in OB 1 and OB 100. You can delete empty runtime groups in the CFC runtime editor via the menu item "Edit > Delete Empty Runtime Groups".

Empty runtime groups are not created in the cyclic interrupts during migration, because they are converted during migration.

- Deleting a CFC
  If a CFC with activated chart-based runtime group management is deleted, all blocks and all the associated runtime groups are deleted as well.
- Renaming a CFC
  If a CFC with activated chart-based runtime group management is renamed, all associated runtime groups are automatically renamed as well. All the names of the associated runtime groups are formed by the CFC name and an appended number for this.

Removing CFCs from the chart-based runtime group management

Chart-based runtime group management can also be deactivated for CFCs in which it is activated.

The menu command "Charts > Chart-based runtime group management" is available for this purpose in the SIMATIC Manager from the shortcut menu of the chart folder or individual charts.

The chart-based runtime group management can be deactivated with this command for a chart folder or individual, selected charts.

Note

Before the conversion, note that the chart-based runtime group management is required to use the "Selective download" function for CFCs/SFCs.

Copying CFCs

When copying CFCs in the SIMATIC Manager, all runtime properties of the blocks contained in the charts basically remain unchanged.

- When a CFC with chart-based runtime group management is copied, the chart-based runtime group management remains activated for the copy of the CFC as well, even if the "Chart-based insertion" option is not selected in the associated chart folder.
- When a CFC with deactivated chart-based runtime group management is copied, this setting also remains deactivated in the copy.

Moving CFCs

In SIMATIC Manager, the behavior when moving CFCs with activated chart-based runtime group management is similar to that described above for copying CFCs.

When moving CFCs in the plant hierarchy, only the assignment to the PH node is changed. The moved CFCs themselves remain unchanged.

Moving blocks between CFCs

- Moving between two CFCs in which chart-based runtime group management is not activated:
  When you move a block between CFCs in which chart-based runtime group management is not activated, only the chart association is changed. The runtime properties of the block, i.e. the runtime groups, are not changed.
- Moving between two CFCs in which chart-based runtime group management is activated:
  The block is installed in the runtime groups which are assigned to the destination chart and removed from the runtime groups of the previous CFC.
- Moving from a CFC without chart-based runtime group management to a CFC with chart-based runtime group management:
  This change is possible.
- Moving from a CFC with activated chart-based runtime group management to a CFC without chart-based runtime group management:
  This change is possible.

Moving and copying runtime groups

- Move
  In CFCs with chart-based runtime group management, the associated runtime groups can only be moved if this is done within a CPU and between two cyclic tasks.
- Copy
  Copying the runtime groups of CFCs with activated chart-based runtime group management is not supported.
  However, the runtime groups of CFCs without chart-based runtime group management can be copied as previously.

Moving and copying block calls between runtime groups

Moving block calls:

- Moving from the runtime group of a CFC with chart-based runtime group management:
  Moving a block call of a runtime group of a CFC with chart-based runtime group management is only possible if the target runtime group belongs to another cyclic task of the same CFC.
  It is possible to move a block call in a runtime group of a CFC with chart-based runtime group management directly below a task. Another runtime group of this CFC is then created automatically.
- Moving between the runtime groups of two CFCs without chart-based runtime group management:
  When you move a block call between the runtime groups of CFCs without chart-based runtime group management, only the runtime properties of the block, i.e. the runtime groups, are changed. The chart association is not changed.

- Moving of the runtime group of a CFC **without** chart-based runtime group management to the shutdown group of a CFC **with** chart-based runtime group management:
  This change is not possible.
- Moving of the runtime group of a CFC **with** chart-based runtime group management to the shutdown group of a CFC **without** chart-based runtime group management:
  This change is not possible because there is no clear relationship between the destination runtime group and a CFC.

Copying block calls:

- Copying block calls of the runtime group of a CFC with chart-based runtime group management is only possible if the block calls are copied to a runtime group of an acyclic task of the same CFC.
  This is only useful if the blocks are to run in another acyclic task that is not included in the task list of the associated block types.
- It is possible to copy block calls between the runtime groups of CFCs without chart-based runtime group management.
- Copying block calls from the runtime group of a CFC **without** chart-based runtime group management to the shutdown group of a CFC **with** chart-based runtime group management and vice versa:
  This change is not possible.

See also

Selective download of individual charts (Page 378)

10.11.10 Runtime attributes

Attributes

A runtime group is assigned the following attributes:

- Enable Attribute (Page 444)
- Reduction ratio (Page 450)
- Phase offset (Page 445)

The runtime group passes these attributes on to all the objects it contains.

The attributes are assigned when the runtime group is created. The reduction ratio and phase offset can also be modified later in the object properties. The enable attribute can be controlled via an interconnection.

NOTICE
Using reduction ratios and phase offsets Whenever possible, only use the reduction ratio and phase offset in tasks that run in defined cycles, in other words in cyclic interrupts. Handle all other tasks very carefully, particularly the hardware interrupt OBs and any special OBs. Here, you should not change the default reduction ratio=1 and phase offset=0.

10.11.11 Display of block execution

Introduction

In CFC, blocks that are not executed are identified visually. This is done in the runtime properties field in the block header.

Edit mode

In edit mode (offline), the blocks are identified as follows:

- Block is executed (no identification)

  The following blocks are identified in this way:

  – Blocks whose EN input is not interconnected and is set statically to 1.

  – Blocks which are located in runtime groups that are not interconnected and are activated statically.

  – Blocks for which there are no SFC write accesses to their block EN and their runtime group EN.

- Block is not executed (red field with an exclamation point)

  The following blocks are identified in this way:

  – Blocks whose EN input is not interconnected and is set statically to 0.

  – Blocks which are located in runtime groups that are not interconnected and are deactivated statically

  – In test mode: If the interconnection to the EN block or EN runtime groups writes a "0".

- Block execution uncertain (yellow field with a question mark)

  The following blocks are identified in this way:

  – Blocks that are not covered by the above lists, for example because their EN is interconnected or SFC access is in place (or if their runtime group is interconnected or SFC writing operations exist there), and whose execution status is therefore not static.

Test mode

In test mode (online), the CFC obtains the enable values of the blocks and runtime groups and uses them to control the display. Here, there are two states:

- Block is executed

  (no identification) if the EN of the block and the EN of the group are set to "1".

- Block is not executed

  (red field with an exclamation point) if the EN of the block or the EN of the group is set to "0".

10.11.12 Handling in CFC

10.11.12.1 Handling blocks in the CFC editor

Handling blocks in unmanaged charts in the CFC editor

- **Copy/Instantiate** (from library)
  Block installation pointer determines the main insertion position. It is set to the inserted block or to the most recently inserted block in the case of several blocks.
- **Move** (only possible within the same CPU)
  Block will be moved to a new chart. The run sequence does not change.
- **Delete**
  Block installation pointer is set to the block preceded in the run sequence.

Handling blocks in managed charts in the CFC editor

- **Copy**
  Block installation pointer is not taken into account. Blocks are inserted in the standard runtime group after the most recent previous block in the run sequence.
- **Instantiate** (from library)
  Block installation pointer determines the main insertion position. It is set to the inserted block.
- **Move** (only possible within the same CPU)
  Block is moved to the target chart and inserted in the standard runtime group. The block is removed from the source chart.
- **Delete**
  Block installation pointer is set to the block preceded in the run sequence.

10.11.12.2 Handling blocks in the runtime editor

Handling blocks in unmanaged charts in the runtime editor

- Copy
  Block is inserted at the selected destination.
- Move
  Block is inserted at the selected destination. If the block installation pointer was set to the moved block in the source chart, it will be moved to the predecessor.
- Delete
  Block instance is deleted if there is another insertion. The last insertion cannot be deleted. The block installation pointer is set to the predecessor.

Handling blocks in managed charts in the runtime editor

- Copy
  A double insertion in the cyclic OB will be rejected. Additional insertions in non-cyclic OBs (for example, OB100) are possible.
- Move
  Block is inserted at the destination only in runtime groups of the same chart. If OB is the destination, a new runtime group is created. Target destination outside the chart will be rejected.
- Delete
  Block instance is deleted if there is another insertion. The last insertion cannot be deleted.

10.11.12.3 Handling charts in SIMATIC Manager

Handling unmanaged charts in SIMATIC Manager

- Copy/Move (without conflict at destination)
  Insertion at the selected destination takes place as specified in the source (graphic and name of the runtime groups). In the case of naming conflict, the runtime group name is automatically resolved by the addition of the suffix "(x)" (where x is 1, 2, 3, 4, and so on).
  Internal interconnections are retained and external interconnections (inputs) are transformed to textual interconnections in the case of an inter-AS action.
  The properties of the runtime group are copied.
  Interconnections to a runtime group are lost.
- Copy/Move (with "Overwrite")
  Insertion at the selected destination takes place as specified in the source (graphic and name of the runtime groups). In the case of naming conflict, the runtime group name is automatically resolved by the addition of the suffix "(x)" (where x is 1, 2, 3, 4, and so on).
  Internal interconnections are retained and external interconnections (inputs) are transformed to textual interconnections in the case of an inter-AS action.
  The properties of the runtime group are copied.

Handling managed charts in SIMATIC Manager

- **Copy/Move** (without conflict at destination)
  Insertion at the selected destination takes place as specified in the source (graphic and name of the runtime groups). In the case of naming conflict, the runtime group name is automatically resolved by the addition of the suffix "(x)" (where x is 1, 2, 3, 4, and so on).
  Internal interconnections are retained and external interconnections (inputs) are transformed to textual interconnections in the case of an inter-AS action.
  The properties of the runtime group are copied.
  Names of the runtime groups are not copied.
- **Copy/Move** (with "Overwrite")
  Insertion at the selected destination takes place as specified in the source (graphic and name of the runtime groups). In the case of naming conflict, the runtime group name is automatically resolved by the addition of the suffix "(x)" (where x is 1, 2, 3, 4, and so on).
  Internal interconnections are retained and external interconnections (inputs) are transformed to textual interconnections in the case of an inter-AS action.
  The previously existing external interconnections in the destination are lost.

When overwriting a hierarchical chart in the CFC editor, the external interconnections will be retained. The parameters perform as set in the settings (parameter setting of a hierarchical chart).

Note

In the CFC editor, you can use the "Copy/Move" menu command under the menu entry "Options" > "Customize" to open the "Settings for Copying/Moving“ dialog box. This dialog box enables you to control the behavior when copying/moving blocks and charts.

Technological configuration 11

11.1 Configuring and managing control modules

11.1.1 Basic information on control modules and their types

Introduction

You can configure control module types and control modules in the same way as process tag types and process tags. Control modules and their types offer the following advantages:

- Synchronization is available when there are differences between the type and instance. Instance-specific extensions are managed as such and are not lost when the type and instance are synchronized. Changes can therefore be bumplessly loaded in the automation system.
  You can find more information on synchronization in the section "Relevant attributes for synchronization of control module types and their instances (Page 223)".
- A control module type can include optional blocks. If different instances of a control module type are created, the optional blocks can be inserted into different instances in different versions.
- The use of control module types and control modules enables extended data exchange with Advanced ES and COMOS Integrated Engineering.

Components of a control module type

A control module type consists of:

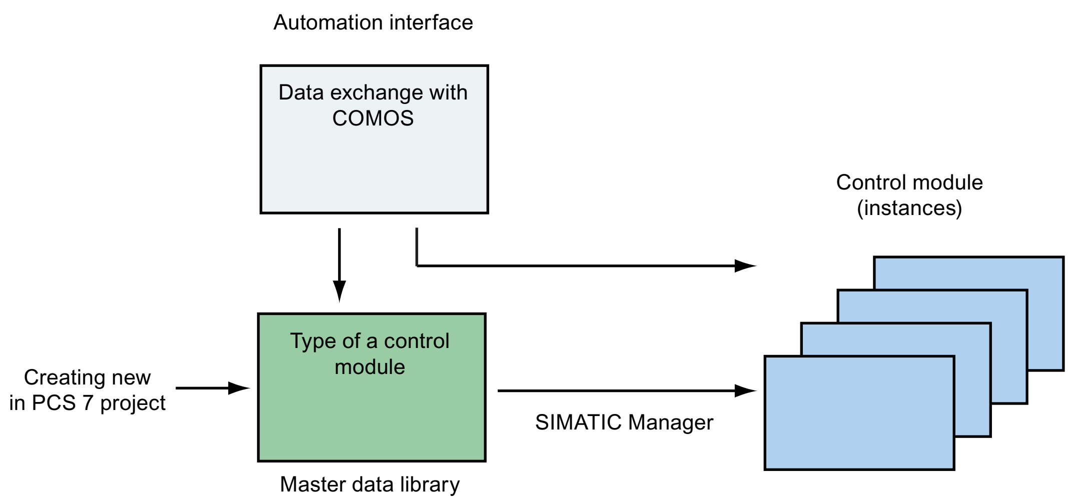
- the technological interface.
  This can optionally contain the following objects:
  - Sub control modules; possibly more than one
  - Control variables; including possibly more than one of the following objects:
    Parameters
    Signals
  - Messages; several are possible
  - Commands; several are possible
  - Statuses; possibly more than one
- The assigned CFC with its possible components, for example, blocks, subcharts.

You can find additional information on this in the section "Overview of data objects of the equipment module and control module (Page 269)".

Creating a type and instance for a control module

You can create control module types from the plant view of a project library in SIMATIC Manager or directly in the CFC editor. They can also be converted from a process tag type.

The figure below provides you with an overview of the creation of a control module type as a new module type or by migration from a process tag type and the creation of associated instances.



Naming the type and instance

- The name of a control module type within the master data library must be just as unique as the types of equipment module.
- The types of control modules share a namespace with the following objects in the master data library:
  - Type of an equipment module.
  - Global commands and status

  A control module therefore cannot have the same name as an equipment module type.
- Name of a control module type:
  The maximum permitted length of the name is 22 characters.
  It is recommended to use only 18 characters to provide some reserve for automatic renaming, for example, by the automatic addition of the character string "(1)" to avoid name collisions during copying.

Note

If the name already exists during copying:

If you copy a chart and paste it, the chart name suffixed with "(x)" (where x is 1, 2, 3, 4, and so on) is assigned as the name of the copied chart.

For example:

If there exists a chart named "ChartName" and if you create a copy of this chart, the copied chart is named as "ChartName(1)" because the name "ChartName" already exists. Further, if you paste it again (create another copy), the next copied chart is named as "ChartName(2)". But if you create a copy from the "ChartName(2)" chart then the copied chart will be named as "ChartName(2)(1)".

Note

If the maximum length of the name is exceeded during copying:

If you copy a chart which is named with 22 characters (maximum length) and paste it, the name of the copied chart is shortened from the end of the string until "~(x)" can be appended (where x is 1, 2, 3, 4, and so on).

For example:

If there exists a chart named "ChartNameMotorControlA" (22 characters) and if you create a copy of this chart then the copied chart is named as "ChartNameMotorCont~(1)".

By the general rules of naming in the case where the name already exists, the new chart name should have been "ChartNameMotorControlA(1)" (25 characters) but this exceeds the maximum permitted length of 22 characters. Therefore, the name is shortened to 22 characters with the methodology explained above.

You can find information on creating a new control module type in the section "How to create a control module type (Page 207)".

You can find information on creating a new control module instance in the section "How to create a control module instance (Page 210)".

Information on creating control module types using migration can be found in the "Process tag type migrator help".

Editing of control module types

Control module types and their associated instances are edited in their own window within the CFC Editor - the technological editor.

You can find additional information on this in the section "Editing of control module types (Page 208)" and "Editing of control module instances (Page 211)".

Particularities when copying and moving a control module

- The following particularities apply when moving a control module:

Required action	Action possible?
Moving between two PH hierarchy folders	Yes
Moving from project to the master data library (component view)	No
Moving to another chart folder (component view)	Possible when the move takes place between different CPUs of the same subproject.

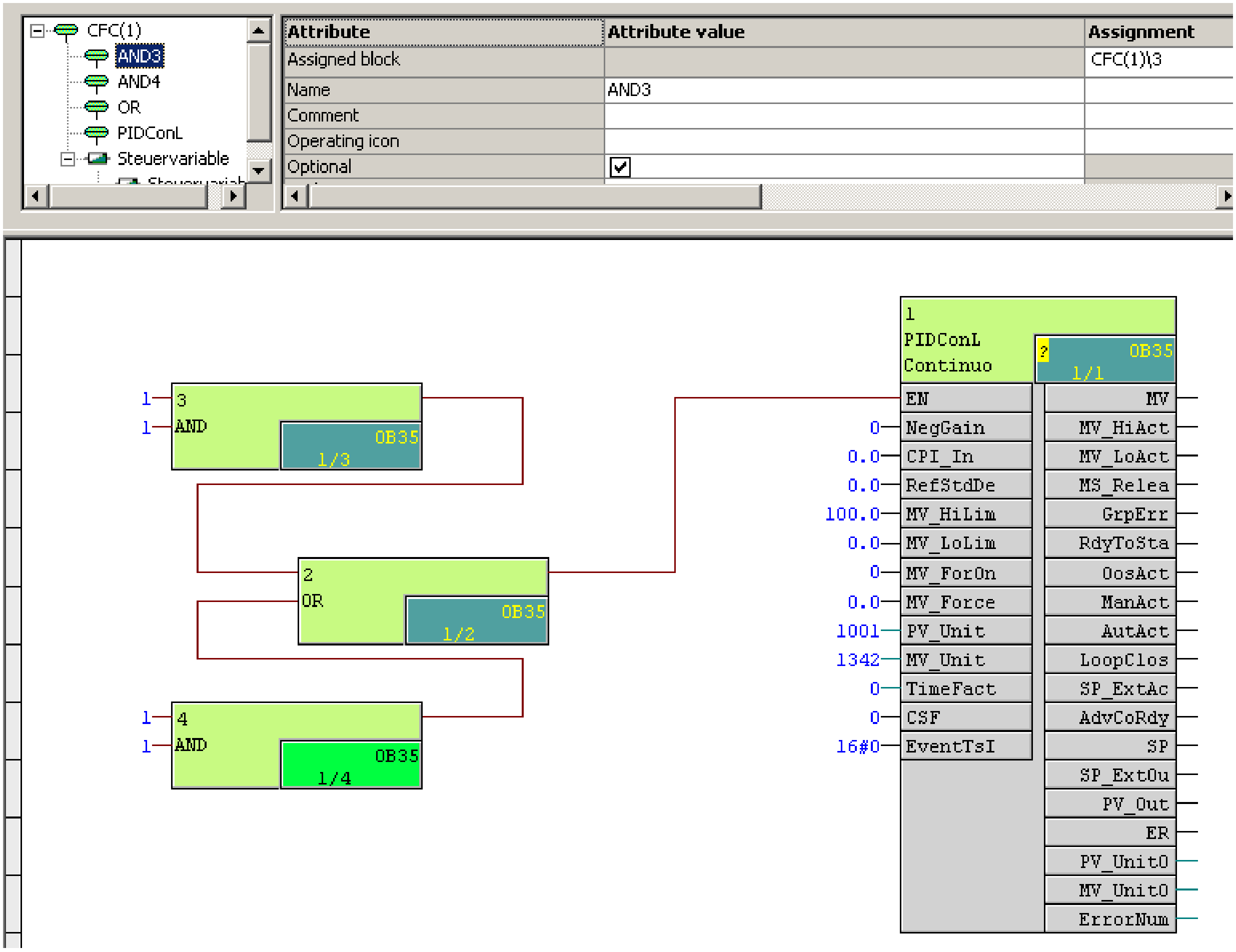
- The following particularities apply when copying a control module:

Required action	Action possible?
Copying between two PH hierarchy folders	Yes
Copying from a PH hierarchy folder to the master data library (plant hierarchy)	Yes Note: A new type is hereby created in the master data library from the instance.

Optional blocks and interconnections on a control module type

Blocks can be defined as optional within a control module type, which means that an individual selection of optional blocks and interconnections can be made for each instance when you create an associated control module.

Example: Control module type in the technological editor and with optional blocks



Additional characteristics of control module types

- You can drag several interconnections to one input within control module types to define alternatives, which is not possible with a CFC. The prerequisite for this is that the "Optional" attribute is activated in the control module type to be configured as interconnection target.
- In the CFC Editor, elements that belong to the definition of the control module type, such as blocks and parameters, are highlighted in color. You may customize the colors used for this purpose.

11.1.2 How to create a control module type

Requirement

You have created a multiproject with a master data library using the SIMATIC Manager.

Procedure

To create a control module type in the SIMATIC Manager, follow these steps:

1. Open the library in the project.
2. Use the menu command **View > Plant view** to change to the plant view
3. Right-click a folder to select the menu command **Insert new object > Control module type**. The control module type is inserted and assigned a default name by the system that you can change to suit your purposes.

Result

The control module type has been created. You can specify the structure of the control module type in a new window and assign the corresponding objects or attributes in the CFC.

You can find additional information on this in the section "Editing of control module types (Page 208)".

11.1.3 Editing of control module types

Requirement

You have created at least one control module type. You can find information on creating a control module type in the section "How to create a control module type (Page 207)".

Editing of control module types within the CFC Editor

If a control module type is initially created or opened, a new editing window opens in the CFC Editor. This technological editor provides the following operating options:

The left section of the technological editor shows the objects of the control module type arranged hierarchically.

Here you can edit the following objects:

- Control module
- As control variables:
  - Signals
  - Parameters
- Message
- Command
- Status

You can create, copy and delete these objects via a shortcut menu. To create such an object, open the shortcut menu with a right-click on the higher-level control module and then select a new object. You can also insert a block from the CFC Editor to the hierarchy of the control module with drag-and-drop.

Note

For configuring messages, refer to the note "Configuration note for messages of a control module or sub-control module" in the section "Overview of data objects of the equipment module and control module (Page 269)".

The attributes and assignments are located in the right section of the technological editor. The attributes that are displayed depend on which object is selected in the left window. Depending on the attribute, you can fill the text fields either with a free text per drag-and-drop from the CFC Editor or from a drop-down menu. Boxes that are grayed out cannot be edited.

You can find more information on attributes in the section "Attributes and attribute values of a control module (Page 214)".

Data types for control variables "Parameter" and "Signal"

A control variable is either of the "Parameter" or "Signal" variable type, depending on its assigned block I/O.

Only the REAL, BOOL, INT and DINT data types are valid for the "Signal" variable type because these define an I/O channel request.

However, the data type at the assigned driver block may also be BYTE or WORD for the raw value. In such a case, the data type is mapped as follows:

- BYTE at the driver block is equivalent to BOOL at the control variable
- WORD at the driver block is equivalent to REAL at the control variable

Exchange of functions in COMOS

You can use the simple functions in COMOS, for example, to configure an interlock logic for control modules. Following data exchange with COMOS, these functions are mapped to control modules. The AND, OR, XOR and NOT functions / standard functions are predefined in PCS 7. They are available on the "Templates" tab of the template catalog, provided a control module has been created.

The user is allowed to define other functions / standard functions as well. This is done by creating a control module that has an interface which exactly corresponds to the definition in COMOS. The names of the control modules and control variables and their data type must match precisely. The check box for the "Function" attribute must be selected. The PCS 7-specific implementation of the function is defined in the corresponding CFC.

Along with the standard data types, functions may be assigned the generic data types ANY_BIT, ANY_NUM, ANY_REAL or ANY_INT. These are displayed in the technological editor by the "Data type" attribute of control variables. The following rules apply if a control variable with a generic data type is assigned to a block I/O in the CFC:

- ANY_BIT is assigned BOOL, BYTE, WORD, or DWORD.
- ANY_NUM is assigned REAL, INT or DINT.

- ANY_REAL is assigned REAL.
- ANY_INT is assigned INT or DINT.

Only block I/Os that fulfill the rules described above can be assigned. The user has to perform the assignment manually.

Display in the CFC Editor

- Control modules are color-coded in the CFC Editor. The following objects are marked:
  – Blocks and interconnections that belong to the control module type
  – Blocks and interconnections that are optional
  – Connections assigned to the control module type
  You may customize the colors used for this purpose. To do so, select "Options > Customize > Colors...". In the next "Color Settings" dialog, select the "Blocks and I/Os of the control module type" object under "Objects".
- Blocks that are part of the standard functions are also color-coded in the CFC Editor.
  You may customize the colors used for this purpose. To do so, select "Options > Customize > Colors...". In the "Color Settings" dialog which follows, select the object "Standard function" under "Objects".
- Block parameters for which the value attribute is ignored for type update are color-coded in the CFC Editor.
  You may customize the colors used for this purpose. To do so, select "Options > Customize > Colors...". In the "Color Settings" dialog which follows, select the object "Ignore value for type update" under "Objects".

11.1.4 How to create a control module instance

Requirement

- You have created a control module type in the master data library of a multiproject.
- You have defined the optional control modules in the control module type.

Procedure

To create a control module instance in the SIMATIC Manager, follow these steps:

1. Open the master data library in the “Plant View”.
2. Right-click a control module type in the hierarchy folder (master data library) and select “Copy” in the context menu.
   Alternatively, select the control module type and press Ctrl+C on your keyboard.
3. Right-click a plant hierarchy folder (project) where you want to create the instance of the copied control module type and select “Paste” in the context menu.
   Alternatively, select the hierarchy folder and press Ctrl+V on your keyboard.

Control module instances can also be created easily dragging and dropping a control module type from the master data library to a plant hierarchy folder (project).

Result

The control module instance is created and assigned a default name by the system that you can change to suit your purposes.

You can further create multiple instances by instantiating the type as well as copying the already existing instances.

You can find additional information on this in the section Editing of control module instances (Page 211).

11.1.5 Editing of control module instances

Requirement

You have created at least one control module instance. You can find information on creating a control module instance in the section How to create a control module instance (Page 210).

Editing of control module instances within the CFC Editor

Control module instances are edited similarly to the control module types. You can find information on editing a control module types in the section Editing of control module types (Page 208).

General behavior of control module instances

Control module instances are derived from their corresponding control module types. Therefore, objects which are derived from control module type cannot be changed in the instance. However, you can additionally insert blocks and interconnections in an instance (generally referred to as “instance-specific blocks” and “instance-specific interconnections”). Objects which are additionally inserted in an instance can be changed.

Changing block parameters of an instance

You can change the block parameters of an instance-specific block.

Block parameters of an instance which is derived from control module type can also be changed if:

- The parameters are part of control module interface (i.e. technological I/Os).
- The parameters are not included in type-instance synchronization (i.e. “Pink” parameters).
- The parameters are relevant for operator control and monitoring (OCM) from the OS (system attribute "S7_m_c" = TRUE).

11.1.6 Configuring a command or status at the type of the control module

Introduction

Commands and status can only be defined at the type of control module, not at an instance.

Commands and status are identified by a name and there can be more than one. They can be selected and copied individually or collectively in the technological editor.

You can find more information about commands and status in the section "Overview of data objects of the equipment module and control module (Page 269)".

Requirement

A control module type is created in the master data library.

Procedure

1. In the master data library, open the control module type in the CFC editor.
2. Select the icon for the control module type in the technological editor.
   
3. Select the "Insert new object" command from the shortcut menu.
   All available objects are shown in the shortcut menu.
4. Select the "Command" or "Status" menu command.
   The selected object is inserted and displayed as an icon.
   – Command
     
   – Status
     

     A lower-level "Parameter" object named "OUT" is created automatically for a new status. This parameter has the BOOL data type and serves as the formal output parameter.
5. Select the icon for the "Command" or "Status" in the technological editor.
   The associated attributes are displayed.
   Configure the attributes, for example, the name, comment and author.
6. Select the icon for the "Command" or "Status" in the technological editor.
   In the shortcut menu, select the menu command "Properties".
   The "Properties" dialog box opens.
   For "Command", the dialog has a similar structure to the dialog for step configuration; for "Status", the dialog has a similar structure to the dialog for the configuration of transition conditions.

7. Configure the desired conditions for the command or status.
   You can also use the "Browse" button to open a selection dialog. All the parameters that can be used, i.e. only the block I/Os that have an identifier "S7_contact = true", are displayed in this dialog box.
   Read the section titled "Special considerations for the implementation of specific commands/statuses (Page 213)" for more information.
8. Select the desired parameter. Click the "Apply" button to apply the selected parameter in the line previously selected of the "Properties" dialog box.
   In the "Properties" dialog box, the next row of conditions is selected automatically and the selection dialog box remains open. If needed, repeat this step to configure additional conditions.
   Click "Close" to close the selection dialog.
9. Click "Close" to close the "Properties" dialog box.
   If required, a prompt for applying the configuration is displayed. Confirm this prompt.

Result

Commands and/or status are defined at the control module type.

Lower-level parameters may also have been inserted for these objects optionally.

A parameter named "OUT" is created automatically at the status.

11.1.7 Special considerations for the implementation of specific commands/statuses

Overview

Introduction

A command/status can be defined as follows:

- At the type of a control module (not at an instance) or at a "Control Module (Basic Requirement)".
  These commands and statuses are specific to these objects.
- "Globally", i.e. with the SIMATIC Manager in the master data library.

Condition for "specific" commands/statuses

The description below for the implementation of "specific" commands and statuses applies under the following conditions:

- Command/status is a lower-level element of the type of a control module
- Command/status is a lower-level element of a "Control Module (Basic Requirement)", whereby this basic requirement was defined in the master data library by setting the attribute "Basic Requirement" in a control module.

Configuration for implementation

If one of these conditions applies, the following implementation is necessary.

- In the basic requirement or control module type which has lower-level commands/statuses, a block type is configured in the "Assigned Block" attribute.
  A block type can be selected in the "Assigned block" attribute in the basic requirement or control module type.
- This means that only block I/Os of this assigned block type which have the ID "S7_contact = true" can be used in the lower-level commands/statuses.

See also

Configuring a command or status at the type of the control module (Page 212)

Overview of data objects of the equipment module and control module (Page 269)

11.1.8 Attributes and attribute values of a control module

Attributes and attribute values

In the left section of the technological editor, the objects in a control module or equipment module are represented hierarchically in a tree structure, for example a "Parameter" or "Message".

The right section lists the attributes for the object that is currently selected in the tree.

Note regarding the "Data exchange" column in all subsequent tables

This column indicates whether an attribute is included in the data exchange with COMOS Integrated Engineering or Advanced ES.

Attributes for higher-level control module

The display of the attributes depends on whether the type or the instance of a control module is opened in the technological editor.

Attribute	Description of attribute values	Data exchange
Assigned chart	Cannot be edited	No
Name	Can be edited Forwarded to associated objects. Gets the name of the assigned chart as default value.	Yes
Comment	Can be edited Forwarded to associated objects.	No
Operating icon	Can be edited Forwarded to the block icon (for WinCC) of a block according to the assignment.	No
Author	Can be edited Not forwarded to associated objects.	No

Attribute	Description of attribute values	Data exchange
Version	Can be edited	Yes
Function identifier	Can be edited Forwarded to associated objects. Control modules (CM) on sensor/actuator/controller-level (process tags) are typically named using standards such as plant identification codes or company-specific codes. The CM names are uniquely defined throughout the plant. Often an abbreviation for the device control is used supplemented by a process tag number, for example, M301, VlvL402, etc. In addition to the CM name (process tag name), the function identifier (FID) allows you to describe the function of a CM using coding standards or company-specific codes. Typical examples are PIC, TIC, GAS702, etc. The function identifier can be an alternative to the CM name assigned to the name attribute of a technological block. In this case, the function identifier will be forwarded to the operator station (OS) if "OCM possible" (Operator Control and Monitoring) option is enabled in the block properties.	No
Location identifier	Can be edited Forwarded to the DOCPRO footer "Code field according to location" of the assigned chart.	No
Sampling time (ms)	Can be edited All blocks of the assigned chart are installed in the OB with the most suitable sampling time. The next higher value will be entered in case of an invalid entry. The runtime editor is adapted accordingly.	Yes
Sampling time of F-program (ms)	Can be edited This attribute is optionally displayed when at least one F-block is used in a CFC. If chart-based runtime group management is activated for the CFC, a separate runtime group is used for the F-program.	Yes
Function	Can be edited only at the type Used to identify a control module as a function.	No
Function name	Displays the name of the function. • Cannot be edited at the type of the control module. When the "Function" attribute is enabled, the value of the "Name" attribute is applied automatically. • Can be edited at the instance.	No
Basic requirement	Can be edited Occurs only at the type. A function cannot be basic requirement.	Yes

Attribute	Description of attribute values	Data exchange
Control module type	Cannot be edited Not forwarded.	No
Support type instance behavior	Can be edited The default setting for this attribute is TRUE for every instance of a control module type. This means that the instance is linked with its type and it is included in the type-instance synchronization. You can exclude instances selectively from the type-instance synchronization by deactivating this attribute. Deactivating this attribute for an instance will not eliminate its internal references and relation to the type. This allows you to edit the instance regardless of its type. You can include instances again in the type-instance synchronization by activating this attribute. Activating this attribute reestablishes the link between instance and its type thus including the instance in type-instance synchronization. Remark: • Deactivate this attribute only in exceptional cases, for example, testing, commissioning, selective, or step-by-step update of instances. • Keeping this attribute deactivated for a longer period might result in inconsistency of the instance with its type. Therefore, check the differences before synchronizing plant types.	Yes

Attributes for a lower-level control module

Attribute	Description of attribute values	Data exchange
Assigned block	Cannot be edited	No
Name	Can be edited Forwarded to associated objects. Gets the default value "Control module".	Yes
Comment	Can be edited Forwarded to associated objects.	No
Operating icon	Can be edited Forwarded to the block icon (for WinCC) of a block according to the assignment.	No
Optional	Select the check box for optional activation of this control module.	Yes
Set as default option	Can be edited only if the attribute "Optional" is activated If this attribute is activated for a control module in a type, then this optional control module (CM variant) is activated by default while creating an instance from the type.	No
Author	Can be edited Not forwarded to associated objects.	No
Function	Can be edited only at the type Used to identify a control module as a function.	No

Attribute	Description of attribute values	Data exchange
Function name	Displays the name of the function. • Cannot be edited at the type of the control module. When the "Function" attribute is enabled, the value of the "Name" attribute is applied automatically. • Can be edited at the instance.	No
Control module type	Cannot be edited Displays the name of the control.	No

Attribute for the "Signal"/"Parameter" objects

Attribute	Description of attribute values	Data exchange
Assigned I/O	Cannot be edited	No
Interconnection to	Can be edited Shows the interconnection.	No
Reference CM parameter	Can be edited Absolute or relative addressing possible. Not relevant for "signal".	Yes
Reference block variable	Can be edited Absolute or relative addressing possible. Not relevant for "signal".	Yes
Reference global variable	Can be edited Absolute or relative addressing possible. Not relevant for "signal".	Yes
Name	Can be edited Not forwarded.	Yes
Comment	Can be edited Not forwarded.	Yes
Signal	Can be edited Right-click the text field to open the symbol table of the CFC using the menu command "New Signal Interconnection". The value is forwarded to the "Interconnection" property of the assigned I/O.	Yes
Value	Can be edited Forwarded to the "Value" property of the assigned I/O. Attribute is unavailable for "Signal".	Yes
Low scale value	Can be edited (at the type) Only relevant for analog signals (input and output signals) of "RE-AL" data type. With binary signals the input fields are locked. Not relevant for "Parameters".	Yes

Attribute	Description of attribute values	Data exchange
High scale value	Can be edited (at the type) Only relevant for analog signals (input and output signals) of "RE-AL" data type. With binary signals the input fields are locked. Not relevant for "Parameters".	Yes
Negation	Select the check box to negate the control variable. The box can only be selected if there is an interconnection.	No
Text 0	Can be edited Forwarded to the "Text 0" property of the assigned I/O. This option is only visible when you are editing an I/O of the data type "BOOL" whose system attribute is set to `S7_string_0`.	Yes
Text 1	Can be edited Forwarded to the "Text 1" property of the assigned I/O. This option is only visible when you are editing an I/O of the data type "BOOL" whose system attribute is set to `S7_string_1`.	Yes
Enumeration	Selection via drop-down menu Forwarded to the "Enumeration" property of the assigned I/O. This option is visible only when the I/O has the system attribute `S7_enum` assigned to it.	Yes
Unit	Selection via drop-down menu For signals: Unit is relevant only for analog signals of "REAL" data type. In addition to the text of the unit, a unique identification number according to the standard "Profile for Process Control Devices" is displayed in square brackets. Forwarded automatically to the unit parameter of the related channel block in CFC program. For parameters: Typically related to independent unit parameter of "INT" data type. Parameter value has to be set preferably to a unique identification number according to the standard "Profile for Process Control Devices". Note that the “Unit” attribute is empty in this case. Alternatively, the unit can be configured by text using the attribute “Unit”. In this case, the “Value” attribute has to be set to 0. Note that conventional parameters can only be configured by text using the attribute “Unit”.	Yes
Operation identifier	Can be edited Forwarded to the "Identifier" property of the assigned I/O. The `s7_shortcut` attribute must be configured for this.	Yes
IO type	Selection via drop-down menu Must match IO type of the assigned I/O.	No
Data type	Selection via drop-down menu Must match data type of the assigned I/O.	Yes

Attribute	Description of attribute values	Data exchange
Tag type	"Signal" or "Parameter" (selection via drop-down menu). Forwarded to the assigned I/O.	Yes
Control module type	Cannot be edited	No

Attributes of the "Message" object

Attribute	Description of attribute values	Data exchange
Assigned message	Cannot be edited	No
Name	Can be edited Not forwarded to associated objects. Gets the default value "Message".	Yes
Message class	Cannot be edited This value is inherited from the "Message class" property of the assigned single message.	No
Priority	Can be edited Only numbers possible. The value "0" is entered in the case of an invalid entry. Forwarded to the "Priority" property of the assigned single message.	Yes
Message identifier	Must be specified manually. If a valid message identifier (e.g. SIG1) is entered, the message class, event and origin attributes are automatically applied as long as the assigned block has a message response.	No
Event	Cannot be edited Inherited from the "Event" property of the assigned single message.	No
Information	Forwarded to the "Infotext" property of the assigned single message.	No
Origin	Forwarded to the "Origin" property of the assigned single message.	No
Single acknowledgment	Can be edited Select this check box if the message should be acknowledged as a single message. The attribute can also be edited in the "PCS 7 message configuration" dialog box.	Yes
With acknowledgment	Can be edited Select this check box if the messages generated should be acknowledged. Depending on whether this check box is selected or not, the "Message class" column in the "PCS 7 message configuration" dialog only displays those classes that can be acknowledged or those that cannot be acknowledged. The attribute can also be edited in the "PCS 7 message configuration" dialog box.	Yes

Attribute	Description of attribute values	Data exchange
Trigger action	Can be edited When the check box is selected, the message triggers the standard "GMsgFunction" which you can edit using the "Global Script" editor of PCS 7. You can find the function under "Standard functions/alarm" in the function browser of the Global Script. The attribute can also be edited in the "PCS 7 message configuration" dialog box.	Yes
OS area	Cannot be edited The area assignment of the message is specified here. If no text is specified or if the area is defined as keyword $$AREA $$, the corresponding attribute of the hierarchy folder is evaluated during transfer of data through OS compilation and stored in the OS message texts. The attribute can be edited in the "PCS 7 message configuration" dialog box.	Yes
Batch ID	Cannot be edited The batch ID for the message is specified here. If the keyword @1%s@ is entered for the batch ID, the corresponding attribute is evaluated during transfer of data through the AS-OS connection configuration and stored in the OS message texts as "Batch name". The attribute can be edited in the "PCS 7 message configuration" dialog box.	Yes

Note

It may not be possible to edit some attributes, because of dependencies to other attributes.

Attributes of the "Command" object

Attribute	Description	Data exchange
Name	Can be edited Not forwarded. Gets the default value "Command".	Yes
Comment	Can be edited Not forwarded.	Yes
Author	Can be edited Not forwarded to associated objects.	Yes

Attributes of the "Status" object

Attribute	Description	Data exchange
Name	Can be edited Not forwarded. Gets the default value "Status".	Yes
Comment	Can be edited Not forwarded.	Yes
Author	Can be edited Not forwarded to associated objects.	Yes

"Assignment" column

Depending on the selected object in the left section of the technological editor, you can assign chart, blocks, block I/Os (parameter/signal), or messages to the attributes of the selected object in the "Assignment" column.

You can assign blocks and parameters by dragging and dropping from the CFC editor. You can assign chart by selecting "Add Chart CM" menu command in the context menu of the selected field.

See also

Attributes of the objects of an equipment module (Page 291)

11.1.9 Comparing and synchronizing the type and instances of a control module

Introduction

The type and instance of a control module are identical when the instance is created.

Synchronization of the type and the instances may be required when changes are made to the type or its associated instances.

Comparison and synchronization

During synchronization, all instances of a type in the project are compared to the corresponding type in the master data library. This shows the differences between the types of the control module and their instances (process engineering level) and between the associated objects, such as CFCs (program level).

Attributes and relations that are relevant for synchronization become part of the comparison result. Attributes and relations that are usually modified at the instance, on the other hand, are excluded from the comparison and are not displayed in the comparison result.

The display of the comparison result is based on the instances of the control modules below the plant hierarchy nodes. In addition to the objects mentioned above, you therefore also see the higher-level objects of the control modules in the comparison dialog box. These serve merely as a navigation or orientation aid.

Note

As of CFC V9.0, a VXM license as of V9.0 is required for type instance synchronization of technological objects, e.g. control modules (CMT).

You can find additional information and notes on synchronization in the section "Notes on type instance synchronization of technological objects (Page 347)".

Included objects

The following lower-level objects are included in the synchronization of the type and instances of control modules on the plant level:

- Sub-control modules
- Control variables:
  - Parameters
  - Signal
- Messages
- Commands and status

Note

Behavior of symbolic texts of binary values during synchronization of control module types

The symbolic texts of binary values (BOOL data type) are only synchronized at the instance if a text is entered for the "Text 0" or "Text 1" attribute at the type at the block parameter of a CFC block, e.g. interlock block. If there is a space or an empty string at the block parameter at the type for these attributes, "Text 0" or "Text 1" of this block parameter is not synchronized.

Procedure

1. Select the project in the plant view.
2. Open the shortcut menu and select the menu command "Plant types > Synchronize...". The "Synchronize plant types" dialog box opens.
3. Select the desired types of control modules or equipment modules as well that you want to compare and synchronize in the left column of the table.
4. Click the "Synchronize..." button to start the comparison.
   Only relevant attributes and relations are included in the comparison. You can find additional information on this in the section "Relevant attributes for synchronization of control module types and their instances (Page 223)".

5. The comparison result is displayed.
   The following objects are highlighted in the comparison result:
   – Objects added at the instance, for example sub-functions, messages, functions and function blocks, are marked as additional objects.
   – Deleted objects.
   – Objects with modified attributes.
6. In the comparison result, select the instances to be synchronized to the corresponding type. Select or clear the check boxes next to objects at the left in the dialog.
7. Start the synchronization with the "Synchronize templates" icon.
   The selected instances are synchronized.
   Instance-specific extensions are retained if they do not apply to objects of the type.

Result

A comparison of the types of control modules and their instances has been performed and selected instances have been synchronized.

You can find more information on type/instance synchronization in the section "Effect of changes to the type/instance synchronization with control modules (Page 225)".

11.1.10 Relevant attributes for synchronization of control module types and their instances

Introduction

An identity check of the type and its instances is performed to synchronize control modules and their types. This check is limited to attributes and relations that may **not** be modified at the instance according to the type instance concept.

Precisely these attributes and relations are compared for the synchronization between type and instances to ensure compatibility and conformity.

Comparison and display of result

Attributes and relations that are relevant for synchronization become part of the comparison result. Attributes and relations that are usually modified at the instance, on the other hand, are excluded from the comparison and are not displayed in the comparison result.

The data transfer dialog is available in PCS 7 to display the comparison result and for synchronization. The actual synchronization process is started in this dialog.

The display of the comparison result is based on the instances of the control modules below the plant hierarchy nodes. In addition to the objects mentioned above, you therefore also see the higher-level objects of the control modules in the comparison dialog box. These serve merely as a navigation or orientation aid.

Overview of attributes and relations relevant for synchronization

The table below lists all objects with their attributes and relations that are to be synchronized:

Object	Relevant attributes	Relevant relations
Control module	-	• Assigned block (block instance or sub-CFC) • Assigned name • Assigned comment • Assigned function identifier • Assigned block icon
Control variable	• Negation (only with interconnected inputs) • Name • Variable type (parameter/signal) • Data type • Comment • Operation identifier • Unit • Enumeration	• Interconnection • Assigned I/O (block variable)
Message	• Name • Message identifier	Assigned message
Block instance	• Name • S7_mes • Block type (function/function block)	Block type (instance of function/function block)
Block variable / block I/O	• inverted (only with interconnected inputs) • Name • Comment • I/O type • Data type • Value • S7_edit • S7_visible • S7_mes • S7_archive • S7_enum • S7_string0 • S7_string1 • S7_shortcut • S7_unit	Interconnection source
Sub-CFC	• Name • Version • Author	-
Interface of the CFC	-	-

Object	Relevant attributes	Relevant relations
Interface parameters of the CFC	• Name • Comment • I/O type • Data type • Value • S7_edit • S7_visible • S7_mes • S7_archive • S7_enum • S7_string0 • S7_string1 • S7_unit	
Block message	• Message identifier • Message type	
Sub-message	Message identifier (sub-message number)	

See also

Basic information on control modules and their types (Page 203)

11.1.11 Effect of changes to the type/instance synchronization with control modules

Overview

Blocks, interconnections, I/Os and messages, for example, can be added, edited or removed from existing and instantiated control modules. These changes can be passed on to the existing, corresponding instances of the control modules by executing a synchronization.

Note

Excluding instances in type-instance synchronization

Instances are not synchronized with their types if “Support type instance behavior” attribute is deactivated for the instances in the technological editor.

For more information, refer to the section "Attributes and attribute values of a control module (Page 214)".

The response during synchronization depends on:

- The type of change (parameter value, interconnection)
- The location of the change (type or instance)
- The attributes of the modified parameters:
  - Parameters in the technological I/Os
  - "S7_m_c" attribute for OS-relevant parameters
  - Other attributes, e.g. Comment, etc.

In particular, block parameters are implicitly excluded from type-instance synchronization if any of the following conditions is true:

- The "S7_m_c" attribute = TRUE.
- The "S7_contact" attribute = TRUE.
- The parameter is in the technological I/Os.
- The parameter is interconnected externally.
- The parameter is managed by the driver generator.
- The parameter is marked as "Pink" parameter (that is, the parameter value is ignored for type update).

Note

Excluding block parameters in type-instance synchronization

You can explicitly exclude various attributes of block parameters from type-instance synchronization (for example, value, comment, identifier, unit, and so on) using the "Type update settings" tab.

For more information, refer to the section ""Properties - Block/Chart" dialog box, "Type update settings" tab (Page 442)".

Note

System behavior of "Light green" and "Pink" parameters

Block parameters which are assigned to the technological interface of a technological type (CM, EM, or EPH) (that is, the block parameters which are in the technological I/O) are highlighted in "Light green" color. Such parameters are typically parameterized for each instance individually and therefore, the value of such parameters is implicitly excluded from type-instance synchronization.

Block parameters which have their value attribute set to be ignored for type-instance synchronization are highlighted in "Pink" color. The value of such parameters is excluded from type-instance synchronization.

In the case where a block parameter is in the technological I/O simultaneously the parameter value is set to be ignored from type updated, the block parameter will be highlighted in "Light green" color. In such cases, both behavioral and functional properties are ensured. Attributes will be propagated from the technological CM parameter to the assigned block parameter and vice versa, if the parameter is highlighted in "Light green" color. The parameter value will be excluded from the type-instance synchronization, if the parameter is highlighted in "Light green" color, in "Pink" color, or simultaneously.

Note

System behavior when synchronizing type and instances of control modules

You can find detailed information on system behavior during synchronization between type and instance of control modules at http://support.automation.siemens.com under the entry ID 99861834.

Legend for the tables below

Action	Result after synchronization	
	Change occurred in the type	Change in the instance (before synchronization)
Brief description of the action or change.	This table cell contains a description of the result in the instance after the synchronization, if a change was made only in the type of the control module.	This table cell contains a description of the effect of the synchronization, if a change was made in the instance before the synchronization, a so-called instance-specific change.

NOTICE

Objects and properties not mentioned

For objects and properties that are not mentioned in this description, it must be verified whether these objects and properties comply with the requirements of the project before a type instance synchronization is performed in a project.

Changes in the CFC of the control module

The following generally applies to the synchronization behavior when inserting new FB and/or FC blocks:

- If newly inserted objects are placed in the main runtime group in the type of the control module, for synchronization they are inserted in the instances directly after the same predecessor blocks as in the type.
- The insertion pointer in the instance is ignored and not changed.

Deviations from this rule are described in the following table.

The following are not synchronized:

- Graphic properties of objects, such as the block position, graphical history of interconnections
- Text fields
- Changes in the block code of an FB/FC. These changes are synchronized by the "Update block types" function.

Action	Result after synchronization	
	Change occurred in the type	**Change in the instance (before synchronization)**
Changing the run sequence	This change is not synchronized. The change must be incorporated in the instances manually. Remark: The various predecessor blocks can be made visible in comparison dialog. Select the "with run sequence" option in the "Synchronize plant types" dialog box when starting the synchronization.	This change is not synchronized. The change must be incorporated in the instances manually. Remark: The various predecessor blocks can be made visible in comparison dialog. Select the "with run sequence" option in the "Synchronize plant types" dialog box when starting the synchronization.
Insert block	Object is inserted in the instance. The object in the instance has the same block as a predecessor as in the type.	Object is retained in the instance. Remark: In principle, no synchronization will take place here, because the change is instance-specific.
Delete block	Object is removed from the instance.	• If the object is in the type: Object is created again in the instance. The object in the instance has the same block as a predecessor as in the type. • Otherwise: Object is removed.
Insert sub-control module	Object is inserted in the instance. The object in the instance has the same block as a predecessor as in the type.	Object is retained in the instance. Remark: In principle, no synchronization will take place here, because the change is instance-specific.
Delete sub-control module	Object is removed from the instance.	• If the object is in the type: Object is created again in the instance. The object in the instance has the same block as a predecessor as in the type. • Otherwise: Object is removed.
Insert sub-chart	Object is inserted in the instance. The object in the instance has the same block as a predecessor as in the type.	Object is retained in the instance. Remark: In principle, no synchronization will take place here, because the change is instance-specific.
Delete sub-chart	Object is removed from the instance.	• If the object is in the type: Object is created again in the instance. The object in the instance has the same block as a predecessor as in the type. • Otherwise: Object is removed.
Changing chart comments	Text in the instance is retained.	Text in the instance is retained.

Note

Restrictions regarding the insertion positions

If an object is newly inserted in the type and placed at the first position in a separate runtime group, the new object in the instance is installed incorrectly after the type/instance synchronization.

If an object is newly inserted in the type, configured as "optional" and placed at any given position in a separate runtime group, the object is incorrectly installed when the variant is activated in the instance.

The insertion positions need to be checked after the synchronization in the instances and adjusted if necessary!

Changes in the technological I/Os

- **Objects in the technological I/Os**

Action	Result after synchronization	
	Change occurred in the type	**Change in the instance (before synchronization)**
Insert block	Object is inserted in the instance.	Object is retained in the instance. Remark: Interconnections of a new object to internal objects are treated as external interconnections for synchronization and do not change.
Delete block	Object is removed from the instance.	• If the object is in the type: Object is created again in the instance. • Otherwise: Object is removed.
Insert parameter	Parameter is in the technological I/Os of the instance.	• If the parameter was inserted in the technological I/Os of the type at the top level: Setting from type is adopted. • If the parameter was inserted in the technological I/Os of the type in a nested object: Setting in the instance is retained.
Delete parameter	Parameter is in the technological I/Os of the instance.	• If the parameter was inserted in the technological I/Os of the type: Parameter is created again in the instance. • Otherwise: Parameter is removed from the technological I/Os.

Action	Result after synchronization	
	Change occurred in the type	**Change in the instance (before synchronization)**
Changing parameter comments	The parameter comment is shown in the synchronization result and the text is adopted by the type at the instance. Even when the comment field is empty in the type, a difference is indicated in the synchronization result.	Instance-specific parameter comments are not possible. The parameter comment is shown in the synchronization result and the text is adopted by the type at the instance.
Insert signal	Signal is in the technological I/Os of the instance.	• If the signal was inserted in the technological I/Os of the type at the top level: Setting from type is adopted. • If the signal was inserted in the technological I/Os of the type in a nested object: Setting in the instance is retained.
Delete signal	Signal is in the technological I/Os of the instance.	• If the signal was inserted in the technological I/Os of the type: Signal is created again in the instance. • Otherwise: Signal is removed from the technological I/Os.
Insert sub-control module	Object is inserted in the instance.	Object is retained in the instance. Remark: Interconnections of a new object to internal objects are treated as external interconnections for synchronization and do not change.
Delete sub-control module	Object is removed from the instance.	• If the object is in the type: Object is created again in the instance. • Otherwise: Object is removed.
Insert sub-chart	Object is inserted in the instance.	Object is retained in the instance. Remark: Interconnections of a new object to internal objects are treated as external interconnections for synchronization and do not change.
Delete sub-chart	Object is removed from the instance.	• If the object is in the type: Object is created again in the instance. • Otherwise: Object is removed.

- **"Optional" attribute in the technological I/Os**

Action	Result after synchronization	
	Change occurred in the type	Change in the instance (before synchronization)
Assign "Optional" attribute for an object in the technological I/Os.	Object remains visible in instance and is activated in variant.	Remark: Cannot be performed in the instance because the dialog is not available.
Remove "Optional" attribute for an object in the technological I/Os.	The object is visible in the instance if has not yet been activated. Object is no longer in the "variant".	Remark: Cannot be performed in the instance because the dialog is not available.
Drag object into the type in the technological I/Os and assign "Optional" attribute to the object.	Object remains visible in instance and is not activated in variant. If the option is activated, then the object in the instance is duplicated, but not correctly installed in the sequence.	Remark: Cannot be performed in the instance because the dialog is not available.

Changing values and interconnections

- **Value change to the input parameter**

Action	Result after synchronization	
	Change occurred in the type	Change in the instance (before synchronization)
Change value of an input parameter in technological I/Os.	Value in the instance is retained.	Value in the instance is retained.
Change value of an input parameter with "S7_m_c" attribute	Value in the instance is retained.	Value in the instance is retained.
Change value of an input parameter with "Ignore value" attribute set ("Pink" parameter)	Value in the instance is retained.	Value in the instance is retained.
Value change to the other parameters	Value of the type is transferred to the instance.	Value of the type is transferred to the instance.

- **Internal or external interconnection**

Action	Result after synchronization	
	Change occurred in the type	Change in the instance (before synchronization)
Add external interconnection to "Parameter" object in the technological I/Os	Remark: Cannot be performed in type.	Interconnection is not changed.
Add external interconnection to "Signal" object in the technological I/Os	Remark: Cannot be performed in type.	Interconnection is not changed.
Delete internal interconnection - with parameters in the technological I/Os	No changes in the instance Interconnection of type is not applied.	No changes in the instance Interconnection is not changed.
Add internal interconnection - if parameter with "S7_m_c" attribute	Interconnection of the type is applied.	Interconnection of the type is applied.
Delete internal interconnection - if parameter with "S7_m_c" attribute	Interconnection in the instance is removed.	Interconnection in the instance is removed.
Add internal interconnection on “Pink” parameter input	No changes in the instance Interconnection of type is not applied.	No changes in the instance Interconnection is not changed.
Delete internal interconnection on “Pink” parameter input	No changes in the instance Interconnection of type is not applied.	No changes in the instance Interconnection is not changed.
Add another internal interconnection	Interconnection of the type is applied.	Interconnection of the type is applied.
Delete another internal interconnection	Interconnection in the instance is removed.	Interconnection in the instance is removed.

Changes to properties / attributes of blocks / block I/Os

Action	Result after synchronization	
	Change occurred in the type	Change in the instance (before synchronization)
Change "enumeration" (block I/O)	• If the "Enumeration" field is empty in the type, the entry in the instance is always retained. • If the "Enumeration" field in the type contains text, it is always transferred to the instance.	• If the "Enumeration" field is empty in the type, the entry in the instance is always retained. • If the "Enumeration" field in the type contains text, it is always transferred to the instance.
Change block group	Text in the instance is retained.	Text in the instance is retained.
Change block comments	Text in the instance is retained.	Text in the instance is retained.
Change "Block icon"	Setting of the type is not transferred to the instance.	The setting in the instance is not changed.

Action	Result after synchronization	
	Change occurred in the type	**Change in the instance (before synchronization)**
Change "OCM possible" (block)	Setting of the type is not transferred to the instance.	The setting in the instance is not changed.
Change "Operator authorization level" (block)	Setting of the type is not transferred to the instance.	The setting in the instance is not changed.
Change "Unit" - at parameter in the technological I/Os	No changes in the instance The unit of the type is not applied.	No changes in the instance The unit is not changed.
Change "Unit" - at parameter with "S7_m_c" attribute	No changes in the instance The unit of the type is not applied.	No changes in the instance The unit is not changed.
Change "Unit" - Other parameters	The unit in the instance is removed.	The unit in the instance is removed.
Change "Identifier" (block I/O)	If the "Identifier" field is empty in the type, the entry in the instance is always retained.	If the "Identifier" field is empty in the type, the entry in the instance is always retained.
Change "MES-relevant" (block)	Setting of the type is transferred to the instance.	Setting is transferred from the type to the instance. (= RESET)
Change the message class of a block	Setting of the type is not transferred to the instance.	Setting in the instance is retained.
Change "Readback enabled" (block)	Setting of the type is transferred to the instance.	Setting is transferred from the type to the instance. (= RESET)
Change "Text 0" or "Text 1" (block I/O)	• If the "Text 0" or "Text 1" field is empty in the type, the corresponding entry in the instance is always retained. • Note about the "empty" state: The "Text 0" and "Text 1" fields at the block are considered "empty" as long as there is no explicit entry in the type of control module at the block. • If the "Text 0" or "Text 1" field in the type contains text, it is always transferred to the instance.	• If the "Text 0" or "Text 1" field is empty in the type, the corresponding entry in the instance is always retained. • Note about the "empty" state: The "Text 0" and "Text 1" fields at the block are considered "empty" as long as there is no explicit entry in the type of control module at the block. • If the "Text 0" or "Text 1" field in the type contains text, it is always transferred to the instance.
Change "Invisible" (block I/O)	• The "visible" setting at the type is transferred to the instance. • The "invisible" setting at the type is not transferred to the instance.	• The "visible" setting at the type is transferred to the instance. • The "invisible" setting at the type is not transferred to the instance. The previous setting at the instance is retained.

See also

"Properties - Input/Output" dialog box (Page 450)

11.1.12 Signal processing for HW signals (IO tags) of control modules

Introduction

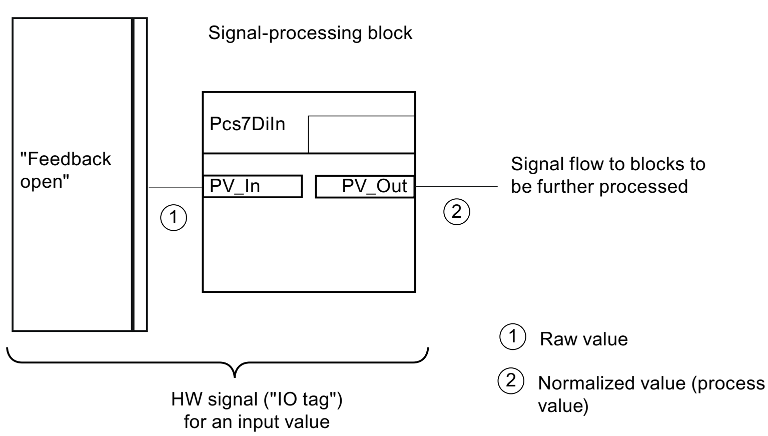
This section provides an overview of the signal processing and interconnection of hardware signals (IO tags) to signals/parameters (CV) in control modules (CM).

Requirements

The following requirements apply for the signal processing in PCS 7:

- Only addresses of the process image are allowed (I, Q) as I/O addresses.
- A "HW signal" (IO tag) must be defined in the CFC editor for each input and output value to be processed:
  - The HW signal (IO tag) is represented in the CFC of a process tag by both the icon in the sheet bar, e.g. "Feedback open", and the signal processing block, e.g. "Pcs7DiIn".
  - The name of the HW signal (IO tag) is determined by the sheet bar entry, e.g. "Feedback open".
  - For input values, the raw value of a HW signal (IO tag) must always be converted to a normalized value with **one** signal processing block, the "Channel driver". The normalized value is then made available at the corresponding output, e.g. "PV_Out", of the signal processing device for further processing.
    Vice versa for output values.
  - Typically, only **one** signal processing block is used for each HW signal (IO tag).

The following figure shows an example of a hardware signal ("IO tag") for an input value.



Overview

Interconnection of a signal (CV) of a control module to a HW signal (IO tag)

An interconnection of a signal (CV) of a control module to a hardware signal (IO tag) can only be made if the associated signal processing block (channel driver) can be determined.

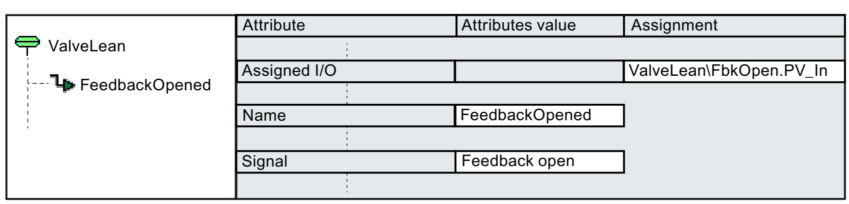
The starting point of the interconnection is always a signal (CV) of a control module. The interconnection target is an input or output of the associated signal processing block. Interconnections of signals can be negated using the "Negation" attribute in the Technological Editor.

The following block inputs or outputs can be connected to a HW signal (IO tag):

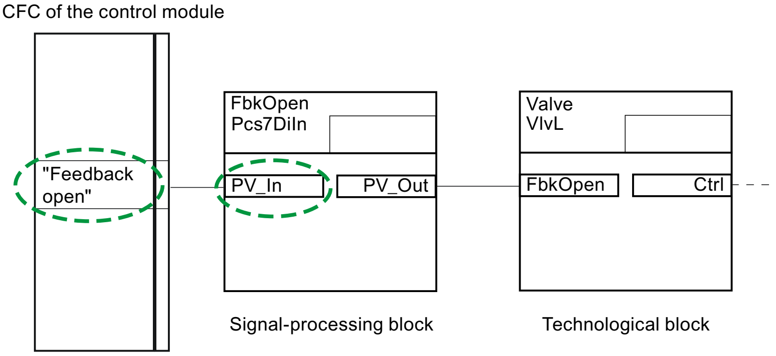
- **Variant 1:**
  Hardware signal assignment to the input of a signal processing block (channel block):
  In this case, the selected hardware signal is assigned directly to the input of a channel block. This interconnection is achieved via signal (CV) in the Technological Editor.
  Configuration:
  – The input of a channel block is assigned to the signal (CV) by dragging-and-dropping the channel block input from the CFC Editor to the “Assignment” field of the “Assigned I/O” attribute in the Technological Editor.
  – The symbolic name of a hardware signal is assigned to the signal (CV) by right-clicking the “Attribute value” field of the “Signal” attribute and selecting a hardware signal in the Technological Editor.

  The signal attribute value of the signal (CV) determines the hardware signal. The I/O assignment of the signal (CV) assigns the hardware signal to the input of the channel block. This establishes an interconnection between the hardware signal and the input of a channel block.
  Example:
  In the Technological Editor, the control module signal (CV) “FeedbackOpened” has the following assignments:
  – Input of the channel block (`FbkOpen.PV_In`) is assigned to the “Assigned I/O” attribute.
  – Hardware signal (symbolic name: `Feedback open`) is assigned to the “Signal” attribute.

Technological editor



CFC of the control module



- **Variant 2:**
  Hardware signal assignment to the input of a technological block (drives, control, or monitoring block):
  In this case, the selected hardware signal is assigned to the input of a technological block by determining the interconnected channel block. This interconnection is achieved via

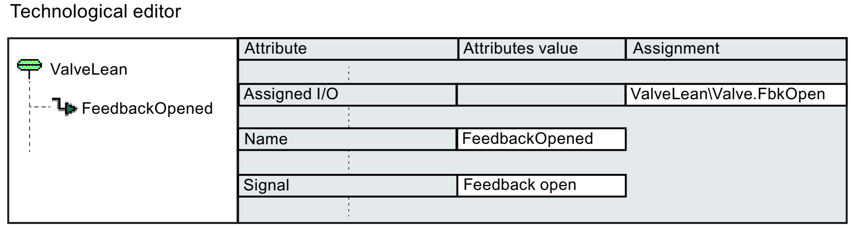
signal (CV) in the Technological Editor and an interconnection between a technological block and a channel block in the CFC Editor.
Configuration:

- An interconnection is made in the CFC Editor between the output of a channel block and the input of a technological block.
- The input of the technological block is assigned to the signal (CV) by dragging-and-dropping the technological block input from the CFC Editor to the “Assignment” field of the “Assigned I/O” attribute in the Technological Editor.
- The symbolic name of a hardware signal is assigned to the signal (CV) by right-clicking the “Attribute value” field of the “Signal” attribute and selecting a hardware signal in the Technological Editor.

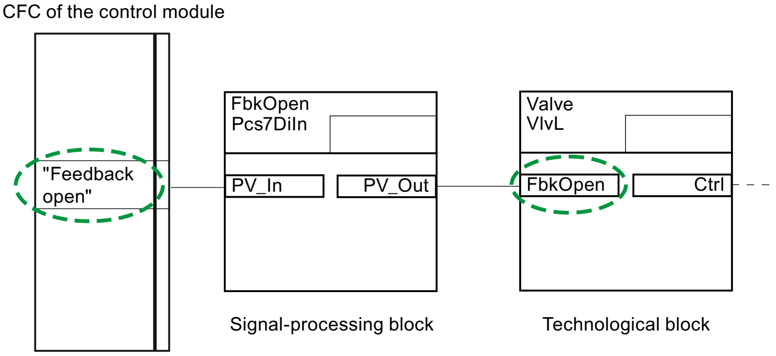
The interconnection between the channel block and the technological block determines the channel block to be used for the hardware signal. The signal attribute value of the signal (CV) determines the hardware signal which will be assigned to the input of the determined channel block. The I/O assignment of the signal (CV) assigns the hardware signal to the input of the technological block. This establishes an interconnection between the hardware signal and the input of a technological block via the determined channel block.
Example:
In the CFC Editor, the output of the channel block (`FbkOpen.PV_Out`) is interconnected to the input of the technological block (`Valve.FbkOpen`).
In the Technological Editor, the control module signal (CV) “FeedbackOpened” has the following assignments:

- Input of the technological block (`Valve.FbkOpen`) is assigned to the “Assigned I/O” attribute.
- Hardware signal (symbolic name: `Feedback open`) is assigned to the “Signal” attribute.

Technological editor

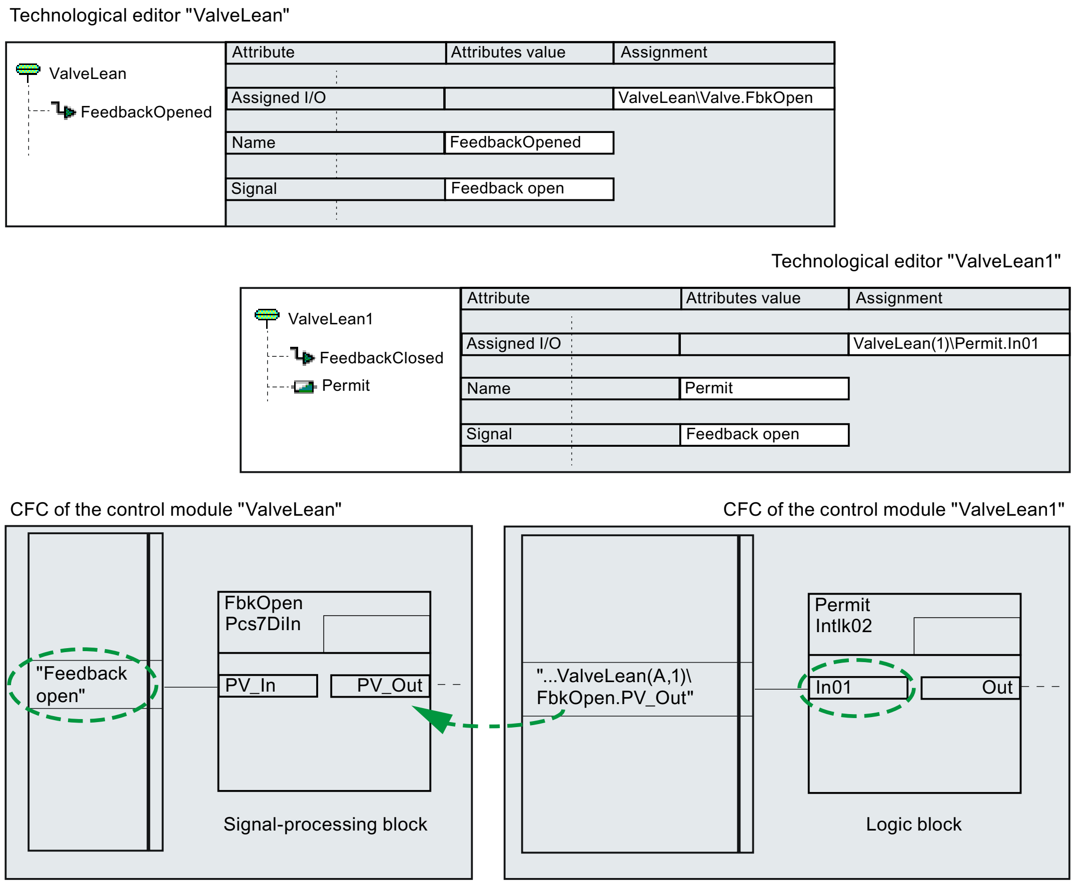


CFC of the control module



- **Variant 3:**
  Hardware signal assignment to the input of a logical block:
  In this case, the selected hardware signal is assigned to the input of a logical block by determining the interconnected channel block. This variant is also used to enable an interconnection to a hardware signal of another control module. The methodology of this variant is same as described in **Variant 2** if the logical block and the channel block are in the same chart. If the logical block and channel block are in different charts, two control modules are needed to make the interconnection.
  Configuration:
  – In the first chart (with a channel block and a technological block), the configuration is same as described in **Variant 2**.
  – In the second chart (with a logical block), the input of the logical block is assigned to the parameter (CV) by dragging-and-dropping the logical block input from the CFC Editor to the “Assignment” field of the “Assigned I/O” attribute in the Technological Editor.
  – The symbolic name of the hardware signal is assigned to the parameter (CV) by right-clicking the “Attribute value” field of the “Signal” attribute and selecting a hardware signal in the Technological Editor.

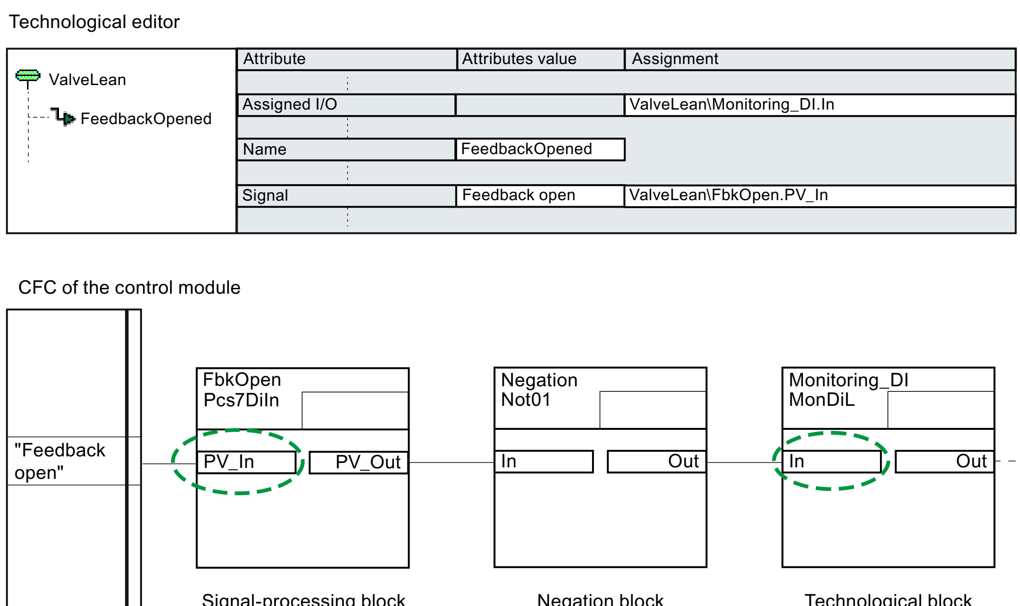
  An automatic interconnection is made between the output of the channel block from the first chart and the input of the logical block from the second chart. This is because, the channel block is already determined in the first chart and the same hardware signal is used in both the charts. Therefore, the second chart (with the logical block) uses the output of the determined channel block from the first chart for interconnection. In the second chart (with the logical block), the I/O assignment of the parameter (CV) assigns the hardware signal to the input of the logical block. This establishes an interconnection between the hardware signal and the input of a logical block via the determined channel block.
  Example:
  In the first chart “ValveLean”, the configuration is same as described in **Variant 2**.
  In the second chart “ValveLean1”, the control module parameter (CV) “Permit” has the following assignments in the Technological Editor:
  – Input of the logical block (`Permit.In01`) is assigned to the “Assigned I/O” attribute.
  – Hardware signal (symbolic name: `Feedback open`) is assigned to the “Signal” attribute.



- **Variant 4:**
  Hardware signal assignment to the input of a technological block in the case where channel block is not connected to technological block directly:
  In this case, the selected hardware signal is assigned to the input of a technological block by inserting the input of a channel block as signal assignment to the input of the technological block. This interconnection is achieved via signal (CV) in the Technological Editor and interconnection between the technological block, an intermediate block, and the channel block in the CFC Editor.
  Configuration:
  – An interconnection is made in the CFC Editor between the output of a channel block and the input of an intermediate block, and then from the output of this block to the input of a technological block.
  – The input of the technological block is assigned to the signal (CV) by dragging-and-dropping the technological block input from the CFC Editor to the “Assignment” field of the “Assigned I/O” attribute in the Technological Editor.
  – The symbolic name of the hardware signal is assigned to the signal (CV) by right-clicking the “Attribute value” field of the “Signal” attribute and selecting a hardware signal in the Technological Editor.
  – The input of the channel block is inserted as signal assignment of the signal (CV) by dragging-and-dropping the input of the channel block from the CFC Editor to the “Assignment” field of the “Signal” attribute in the Technological Editor.

The signal assignment of the signal (CV) determines the channel block. The signal attribute value of the signal (CV) determines the hardware signal which will be assigned to the input of the technological block. The I/O assignment of the signal (CV) assigns the hardware signal to the input of the technological block. This establishes an interconnection between a hardware signal and the input of a technological block directly via the signal assignment of the signal (CV).
Example:
In the CFC Editor, the output of the channel block (`FbkOpen.PV_Out`) is interconnected to the input of the negation block (`Negation.In`) and the output of the negation block (`Negation.Out`) is connected to the input of the monitoring block (`Monitoring_DI.In`). In the Technological Editor, the control module signal (CV) “FeedbackOpened” has the following assignments:

– Input of the technological block (`Monitoring_DI.In`) is assigned to the “Assigned I/O” attribute.
– Hardware signal (symbolic name: `Feedback open`) is assigned to the “Signal” attribute.
– Input of the channel block (`FbkOpen.PV_In`) is inserted as the signal assignment in the “Assignment” field of the “Signal” attribute.



Interconnection rules

There are certain rules are certain rules when interconnecting a signal/parameter (CV) of a single controller to a HW signal (IO tag).

The following interconnections between signals/parameters (CV) of control modules are supported:

- Parameter as input - parameter as output
- Parameter as input - signal as output

The following signals/parameters (CV) can be symbolically assigned to HW signals (IO tags) via the "Signal name" attribute in the Technological Editor:

- Signal as input
- Parameter as input
- Signal as output

Interconnections of the "BOOL" data type or the assignments of digital hardware signals (IO tags) can be negated.

11.1.13 Implementation of signal processing of hardware signals (IO tags) for control modules in XML format

Introduction

Information about the signal processing blocks and the technological blocks is required for signals in control modules to be able to support the interconnections of the signals of a control module.

This information can be provided in XML format.

The following information is defined in XML format:

- Entries for all signal-processing blocks (so-called channel drivers)
  In addition to the block definition, at least one "raw value" - "standardized value" assignment is contained here.
- Entries for technological blocks
  All I/Os of this block that are to have access to the value of an I/O block and therefore require a driver connection are defined here in addition to the block definition.

Provision

- For channel drivers and blocks of the PCS 7 Advanced Process Library (APL), this information is provided in XML format in the "SignalBlocksAPL_8x.xml" file under the "<Installation directory>\\Siemens\STEP7\S7data\SignalProcessing" directory.
- For user-defined driver blocks or those driver blocks that are not included in the PCS 7 Advanced Process Library (APL), this information must be provided by the user as an XML file in the "<Installation directory>\Siemens\STEP7\S7data\SignalProcessing" directory.

The principle of the implementation in XML format is shown below in an example for these user-defined driver blocks.

Principle of implementation in XML

The principle of implementation in XML format is shown based on the provided example configuration with few blocks.

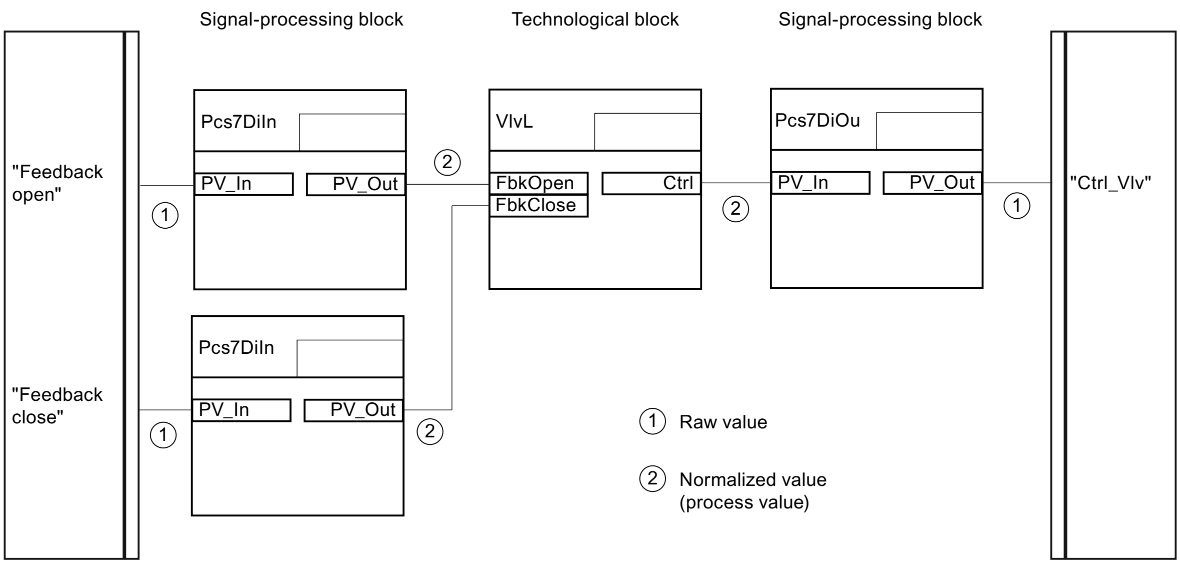
For user-defined driver blocks, the corresponding names, values and types of the example must be replaced with the desired data.

Example configuration

The example configuration only requires a few blocks:

- Pcs7DiIn: Signal processing block of a digital input
- Pcs7DiOu: Signal processing block of a digital output
- VlvL: Technological block for a valve control with I/Os that require a driver connection (channel request)

The following figure shows the blocks in the CFC of the control module and the signal flow.



Definition of the I/Os of the blocks

The following tables show the blocks used in the example and the I/Os involved in the signal flow:

- Signal processing blocks (channel drivers) Pcs7DiIn and Pcs7DiOu
  The table shows, the I/Os for a raw value and normalized value (process value) for each signal processing block.
  The corresponding XML names are shown in parenthesis.

Signal-processing blocks			Raw value (RawValue)		Normalized value (ProcessValue)	
Name (SignalBlock Name)	**Signal type (Signal Type)**	**Data type (DataType)**	**I/O name (RawValue Name)**	**I/O type (Section)**	**I/O name (RawValue Name)**	**I/O type (Section)**
`Pcs7DiIn`	`Input`	`Digital`	`PV_In`	`Input`	`PV_Out`	`Output`
`Pcs7DiOut`	`Output`	`Digital`	`PV_Out`	`Output`	`PV_In`	`Input`

- Signals of the technological block VlvL
  The table shows the I/Os from the example that require a channel connection for the VlvL block.
  The corresponding XML names are shown in parenthesis.

Name of the technological block (ProcessBlock Name)	Signal name (Parameter Name)	I/O type (Section)	Data type (DataType)
VlvL	FbkOpen	Input	Digital
	FbkClose	Input	Digital
	Ctrl	Output	Digital

Implementation of the example configuration in XML

The table shows the XML definitions for the blocks and their I/Os corresponding to the tables above.

```
<?xml version="1.0" encoding="utf-8"?>                                  <!-- Definition begin -->
<SignalBlocks Version="x.y" xmlns="http://www.siemens.com/Simatic/PCS7/SignalBlocks/1.0">
  <SignalBlock Name="Pcs7DiIn">                                         <!-- Signal-processing blocks -->
    <Signal Type="Input" DataType="Digital">
      <RawValue Name="PV_In" Section="Input"/>
      <ProcessValue Name="PV_Out" Section="Output"/>
    </Signal>
  </SignalBlock>
  <SignalBlock Name="Pcs7DiOu">
    <Signal Type="Output" DataType="Digital">
      <RawValue Name="PV_Out" Section="Output"/>
      <ProcessValue Name="PV_In" Section="Input"/>
    </Signal>
  </SignalBlock>
  <ProcessBlock Name="VlvL">                                            <!-- Technological block and its I/Os -->
    <Parameter Name="FbkOpen" Section="Input" DataType="Digital"/>
    <Parameter Name="FbkClose" Section="Input" DataType="Digital"/>
    <Parameter Name="Ctrl" Section="Output" DataType="Digital"/>
  </ProcessBlock>
</SignalBlocks>                                                         <!-- Definition end -->
```

11.1.14 References at the parameters of a control variable

Introduction

So-called technical references are used to be able to exchange an incomplete plant description with established relationships to other process tags with the process control system on the technological level (e.g. in COMOS).

These references can be used in a parameter which is configured as a control variable ("CV") of a control module ("CM") or equipment module ("EM").

No references are possible for signals.

The difference between textual interconnection and reference at a parameter

- A textual interconnection can only exist at one block or chart input and always refers to a block or chart output in the CFC.
  The textual interconnection can only be specified as absolute, not relative.
- The reference at a parameter can reference input and output parameters of other process tags. It is used on the technological level and cannot be mapped to a textual interconnection of the CFC.
  A reference at a parameter can be specified as absolute and relative.

Types of references at a parameter

There are several types of references at a parameter of a control variable, each of which is configured in a corresponding attribute of the parameter:

- Reference to a parameter of a control module / equipment module: "Reference CM parameter" attribute
- Reference to a block variable: "Reference block variable" attribute
- Reference to a global variable: "Reference global variable" attribute

---

Note

"Parameter of a control variable" expression

The following description uses the expression "parameter of a control variable" as shorthand for a parameter that is configured as a control variable of a control module / equipment module. The expression "parameter of a control variable" does refers not an interface parameter of the control module or CFC.

Overview

The following points are described below:

- Configuring a reference at a parameter
- Syntax for references at a parameter
- References for data exchange

- Copying behavior of references
- Completing an open reference:
  - "Reference CM parameter" type
  - "Reference block variable" type
  - "Reference global variable" type

Configuring, changing, deleting a reference at a parameter

- References can be entered manually on the technological level in COMOS and in the technological editor of the CFC based on the syntax described below.
  - To create a new reference in the technological editor of the CFC, the required attribute is selected at the parameter, for example, "Reference block variable". The reference can be entered using the "New Reference" shortcut menu command.
  - Creating a "Reference CM parameter" with drag-and-drop:
    A parameter of another control variable can be dragged with the mouse to the "Reference CM parameter" field of the control variable parameter that is to be configured. A reference with path information to the parameter of the other control variable is automatically created.

    Note

    No names of blocks or block parameters from a CFC can be used to create a "Reference CM parameter" in the technological editor. Instead, the names of blocks or block parameters from the technological editor should be used.

    As mentioned above, it is only possible to configure a "reference CM parameter" with drag-and-drop between the parameters of control variables. The technological editors of the desired control modules / equipment modules must be visible for this.

  - The existing reference can be changed by positioning the cursor at the desired attribute of the parameter in the technological editor of the CFC. This opens a dialog in which the reference can be edited.
  - To delete a references, a command is provided in the shortcut menu of the corresponding attribute of the parameter.
- The addressing of the parameter to be interconnected to a control module / equipment module can also be defined relative to its own hierarchy node.
  Relative addressing is initiated with the character string "..\", the hierarchy of which is based on the parameters of the control module / equipment module with the parameters to which the reference is to be configured. The "..\" string takes the addressing one node higher beginning with this starting point in the plant hierarchy. The "..\" string is repeated until a common hierarchy node is achieved of between the source and destination parameters. Then the addressing starts to go down to the desired target parameter in the hierarchy.
- A reference to a parameter to both the input and on the output end is allowed in contrast to an interconnection.
  - At the input end, a reference can only be configured to either a value or an interconnection.
  - At the output end, multiple references can be configured.
    In this case, references of the various types; "Reference CM parameters", "Reference block variable" and "Reference global variable" can be used in parallel.

- "Reference block variable"
  Typically, interconnections are made between two parameters on the engineering level, e.g. in COMOS. However, to avoid reconfiguration at the PCS 7 end and having to make known all CFC block variables on the technological level, CFC block variables can also be connected directly to a parameter of a control variable. For this, references are configured to block variables at the parameters of a control variable.

  On the technological level in COMOS and in the technological editor of the CFC, references to the CFC block variables can be entered directly according to PCS 7 syntax including the path of the plant hierarchy.
  The input is typically made already on the engineering level, e.g. COMOS.
  The following options are available to configure the "Reference block variable" in the technological editor of the CFC:

  - Manual entry:
    Using the plant hierarchy, a reference to the actual parameters at a control module / equipment module is entered as an absolute project path.
    A reference entered manually is checked neither for syntax nor semantics and is saved unverified!
    There is an automatic attempt to close the new reference to a block variable and thereby make it into a fixed interconnection at the CFC block.

    Note

    If a new reference can be closed and points to a block variable that is already assigned a textual interconnection, the existing textual interconnection at the block variable is replaced by the new reference!

  - Dragging-and-dropping a block parameter:
    A block parameter from the CFC can be dragged with the mouse to the "Reference block variable" field of a control variable parameter. A reference to this block parameter is created automatically.
  -

    A "Reference block variable" can be configured at the input end only as an alternative to a value, an interconnection or another type of reference.
  - A "Reference block variable" is allowed multiple times at the output end of the control module / equipment module.

- "Reference global variable"
  This type of reference provides the ability to configure a connection to global data pools (S7 shared DB).
  On the technological level in COMOS and in the technological editor of the CFC, references to elements of a shared data block can be entered within the context of the programming model of STEP 7. The references can be entered symbolically or per address based on STEP 7 syntax.
  - A reference to block variable can be configured at the input end only as an alternative to a value, an interconnection or another type of reference.
  - A reference to global variable is also allowed multiple times at the output end of the control module / equipment module.
  - The reference from a parameter of a control variable to an element of a data pool is mapped directly to an S7 address interconnection at the associated CFC block I/O.

  If the assignment of the parameter to be interconnected to a CFC block I/O is only made later, the entries relating to the references to the data pools are then automatically transferred as S7 address interconnections. The assignment is rejected if one or more references to a data pool, for example, could not be applied due to a data type incompatibility.

Syntax for references at a parameter

- References are possible for both input and output parameters of a control module / equipment module on a technological level.
- Using the plant hierarchy, a reference to the actual parameters at a control module / equipment module is entered either as an absolute project path or a relative one.
- The following syntax is used for references at parameters.
  Relative paths are preceded by a sequence of the "..\" string.
  - ```
    ( NameOfProject\ { NameOfPlantHierarchyFolder\ } { NameOfCM\ }
    NameOfCM.NameOfCV ) | ( '..\' { '..\' }
    { NameOfPlantHierarchyFolder\ } { NameOfCM\ }
    NameOfCM.NameOfCV )
    ```
    ```

- If the reference addresses a parameter of an equipment module (EM) or the equipment phase (EPH), the following syntax is used:
  - `(NameOfProject\ { NameOfPlantHierarchyFolder\ } { NameOfCM\ | NameOfEM\ } (NameOfCM.NameOfCV | NameOfEM.NameOfCV | NameOfEM \NameOfEPH\NameOfEPHParameter.NameOfCV)) | ('..\' { '..\' } { NameOfPlantHierarchyFolder\ } { NameOfCM\ | NameOfEM\ } (NameOfCM.NameOfCV | NameOfEM.NameOfCV | NameOfEM \NameOfEPH\NameOfEPHParameter.NameOfCV))`
- The following syntax is used for references to block variables:
  - `NameOfProject\\ { NameOfPlantHierarchyFolder\ } NameOfPlantHierarchyFolder\\ ( NameOfCFC.NameOfCFCInterfaceVariable | ( NameOfCFC\ { NameOfSubCFC\ } NameOfSubCFC.NameOfSubCFCInterfaceVariable | NameOfBlock.NameOfBlockVariable ) )`

References for data exchange

- References to a parameter of a control module / equipment module are exchanged bidirectionally during the data exchange between PCS 7 and COMOS Integrated Engineering using the control variable in question.
- Data exchange via XML file:
  References that are configured directly at a parameter of a control module / equipment module or created implicitly with the copy operation are entered in the XML file.
- No references are generated automatically for the data exchange.
- When transferring from COMOS Integrated Engineering or using an XML file, references in PCS 7 are stored at the control variables of the control module / equipment module. This makes them visible as attributes at the parameters of the control variable in the technological editor.
- An assignment at the parameter or an existing interconnection is deleted when the reference is transferred because a reference can only be saved as an alternative.
- When exchanging data and comparing it in the data transfer dialog of the Automation Interface (AI), an open reference is not closed, even if the interconnection partner entered in the reference of a parameter exists.
- The log for the data exchange contains an entry if the parameter is not assigned to a CFC block I/O.

Copying behavior of references

- Only when copying "Reference CM parameter" type references is there an attempt to maintain all interconnections between parameters of control variables configured on the technological level.
  For example, if a control module / equipment module is copied with its CFC implementation in the SIMATIC Manager, then the relative reference is determined for interconnections of control variables both at the inputs as well as the outputs and stored at the control variables of the control module / equipment module copy.
- The copying behavior is independent of the copy destination.
- The transfer or creation of references during a copy operation is enabled by default, but can be optionally switched off.
  An option is available for this in the "Options > Charts > Settings for Copying/Moving" menu of the SIMATIC Manager.

Completing an open reference of the "Reference CM parameter" type

-
  When transferring from COMOS Integrated Engineering or using an XML file, references in PCS 7 are stored at the control variables of the control module / equipment module. This makes them visible as attributes at the parameters of the control variable in the technological editor.
- Each references can be closed at any time manually in the technological editor or with a collective instruction in the SIMATIC Manager.
  The menu command "Options > Plant types > Close references to CM parameters" is provided for this for a selected project in the plant view of the SIMATIC Manager. This menu command is also displayed in the shortcut menu of the project.
- References can be closed only when a connection can be made from one parameter of a control variable to the parameter of another control variable.
- The reference at the parameter is made when the closing operation was successful.
-
  If the reference cannot be closed, corresponding messages are entered in a log.
- Log:
  The log lists successfully completed references and reference that remained opened at this point in time. The log can be accessed with the menu command "Options > Plant types > Logs". This menu command is also displayed in the shortcut menu of the project.

Completing an open reference of the "Reference block variable" type

-
  When transferring from COMOS Integrated Engineering or using an XML file, references to block variables are stored at the control variables of the control module / equipment module in PCS 7. This makes them visible as attributes at the parameters of the control variable in the technological editor.
- When exporting from COMOS to PCS 7, an attempt is made to resolve these references and mapped them to a CFC interconnection.

- Each references that could not be closed after a data exchange from COMOS to PCS 7 or after manual entry in the technological editor of the CFC can be closed manually.
  The menu command "Options > Plant types > Close references to block variables" is provided for this for a selected project in the plant view of the SIMATIC Manager. This menu command is also displayed in the shortcut menu of the project.
-
  If the reference cannot be closed, corresponding messages are entered in a log. One possible cause, for example, is an existing interconnection to another CFC block at the associated block input.
- Log:
  The log lists successfully completed references and reference that remained opened at this point in time. The log can be accessed with the menu command "Options > Plant types > Logs". This menu command is also displayed in the shortcut menu of the project.
- References to block variables are retained even after the closing process, regardless of whether or not a module interconnection could be created.
- At the COMOS end, references to block variables generally remain "open" because on the engineering level, the PCS 7-specific block variables are not known by definition. In other words, only the text of the reference is stored at the attribute of the relevant parameters of the control variables.

Completing an open reference of the "Reference global variable" type

-

  When transferring from COMOS Integrated Engineering or via an XML file, a reference to a global variable in PCS 7 is mapped to an address interconnection at the associated block I/O in the CFC.

  If the reference cannot be closed, corresponding messages are entered in a log. The log can be accessed with the menu command "Options > Plant types > Logs".
- If the parameter is not assigned to a CFC block I/O, this is also reported in the log, and the reference to the global data pool is still stored at the control variables (CV) of the control module.
- An assignment at the parameter or an existing interconnection is deleted when the reference is transferred because the reference can only be saved as an alternative.
- However, an S7 address interconnection can remain "open" in respect to its address, when the global DB has not yet been fully created and or the symbolic name of the global DB has not yet been entered in the symbol table.
  In these situations, the user must subsequently compare the S7 address interconnections with the symbol table and the global DB. The references to global data are defined CPU-wide because of their PCS 7 restriction.
  Synchronization is performed automatically at the latest during the build by the S7 compiler. When a reference to a global variable in the global DB cannot be resolved, this is reported in the log of the S7 compiler.
  Manual synchronization with the symbol table can be started with the key combination "Ctrl + F5" in the CFC at any time.
- References to global variables remain open at the COMOS end. In other words, only the text of the reference is stored at the attribute of the parameter of the control variables.

11.1.15 Configuration examples for interconnections of optional blocks

Introduction

A control module can contain "optional" blocks.

These optional blocks can only be defined as "optional" at the control module type in the master data library.

If different instances of this type are created in the project, you can use the "variant selection" to determine for each instance which optional blocks are contained in this instance.

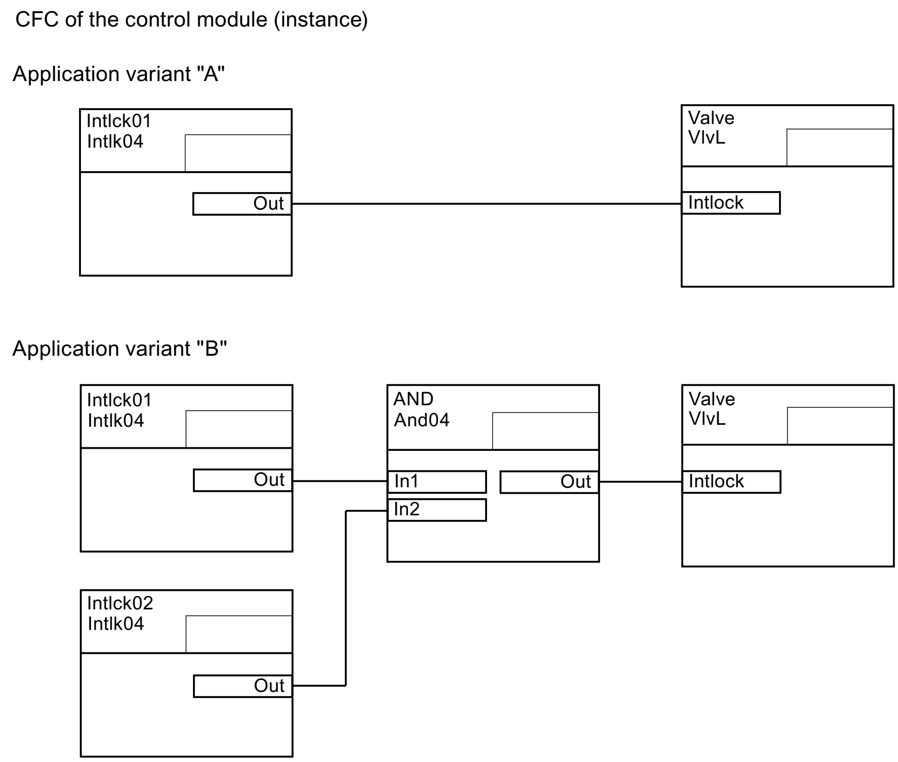
Below you will find a description of configuration examples for interconnections between optional and non-optional blocks. The examples also show the effect of the selection of optional blocks on the interconnections.

Application variants

The figure below shows the application variants that should be available for selection in the type instances by using optional blocks in a control module type.

- Application variant "A"
  In this application variant, the CFC contains the target block "Valve" and only one "Intlck01" block.
- Application variant "B"
  This application variant contains the block "Valve" and two blocks "Intlck01" and "Intlck02".

To interconnect the output signals of the "Intlck" blocks to the "Intlck" input of the "Valve" block, the logic block "AND" is added as well.



Requirement:

The inputs and outputs of the optional blocks required in the application variants must be configured and interconnected as "Parameters" in the technological editor of the control module type.

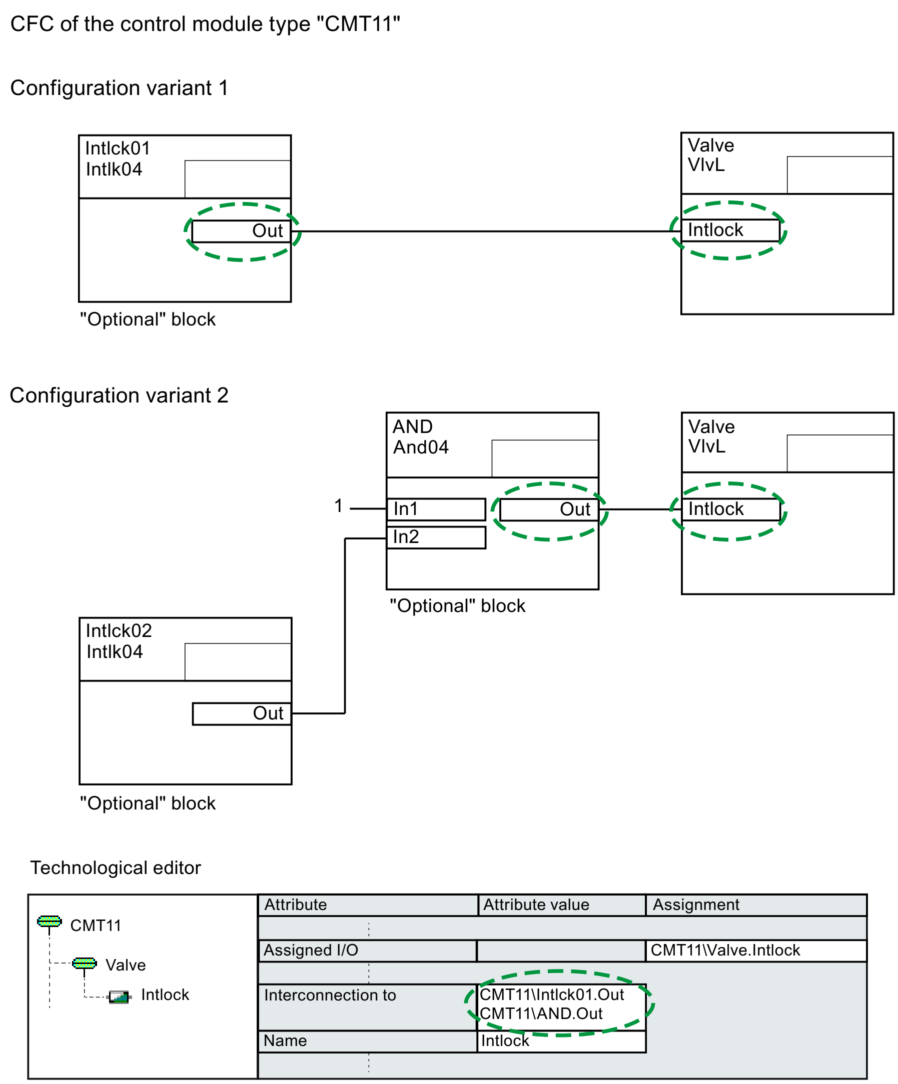
Configuration example 1

In this configuration example, the optional blocks are configured as follows in the control module type:

- In configuration variant 1, the block "Intlck01" is defined as optional.
- In configuration variant 2, the blocks "Intlck02" and "AND" are defined as optional.
- The "Valve" block is not optional.

Variant selection at the instance:

- For the application variant "A" with one "Intlck" block, only configuration variant 1 is activated in the instance.
- For application variant "B" with multiple "Intlck" blocks, both configuration variants 1 and 2 are activated in the instance.



Properties of this configuration example

Configuration:

For this configuration method, the interconnection targets for the two possible application variants are entered at the "Intlock" parameter of the sub-control module "Valve" below the "Interconnection to" attribute in the technological editor.

- For application variant "A" with only one "Intlck" block: "CMT11\Intlck01.Out"
- For application variant "B" with multiple "Intlck" blocks: "CMT11\AND.Out"

Layout of the structure in the technological editor:

The following table shows the associated structure in the technological editor. A sub-control module "Option_x" is created for each of the configuration variants 1 and 2. The inputs and outputs of the optional blocks that are to be interconnected are subordinated as parameters.

Sym-bol	Name/ symbol	Name/ symbol	Parameter name	Remark
	"CMT11"			Control module for configuration example 1
		"Option_1"		Sub-control module for parameters of configuration variant 1
			"Intlck01_Out"	Parameter for "Intlck01" block, "Out" output
		"Option_2"		Sub-control module for parameters of configuration variant 2
			"Intlck02_Out"	Parameter for "Intlck02" block, "Out" output
			"And_Out"	Parameter for "AND" block, "Out" output
		"Valve"		Sub-control module for the "Valve" block as interconnection target
			"Intlock"	Parameter for "Valve" block, "Intlck" input

Effects on the variant selection at the instance:

If the optional blocks for the application variants are activated at an instance of this control module type "CMT11", the chronological order of the activation is important. Only then can the automatic interconnection for the "Intlck" input of the "Valve" block take place correctly.

The reason for this is that only one interconnection can be present at a block input.

If configuration variant 1 is activated first with the optional "Intlck01" block, an interconnection is created between the "Out" output of the "Intlck01" block and the "Intlck" input of the "Valve" block. When activating configuration variant 2 afterwards, the configured interconnection from the "Intlck" input of the "Valve" block to the "Out" output of the "AND" block can no longer be created.

For configuration variant "B", you therefore first have to activate configuration variant 2 with the optional blocks "Intlck02" and "AND". In doing so, the "Intlck" input of the "Valve" block is automatically interconnected to the "Out" output of the "AND" block. Only then is configuration variant 1 activated with the optional "Intlck01" block and the "Out" output of the "Intlck01" block is interconnected to the "In1" input of the "AND" block.

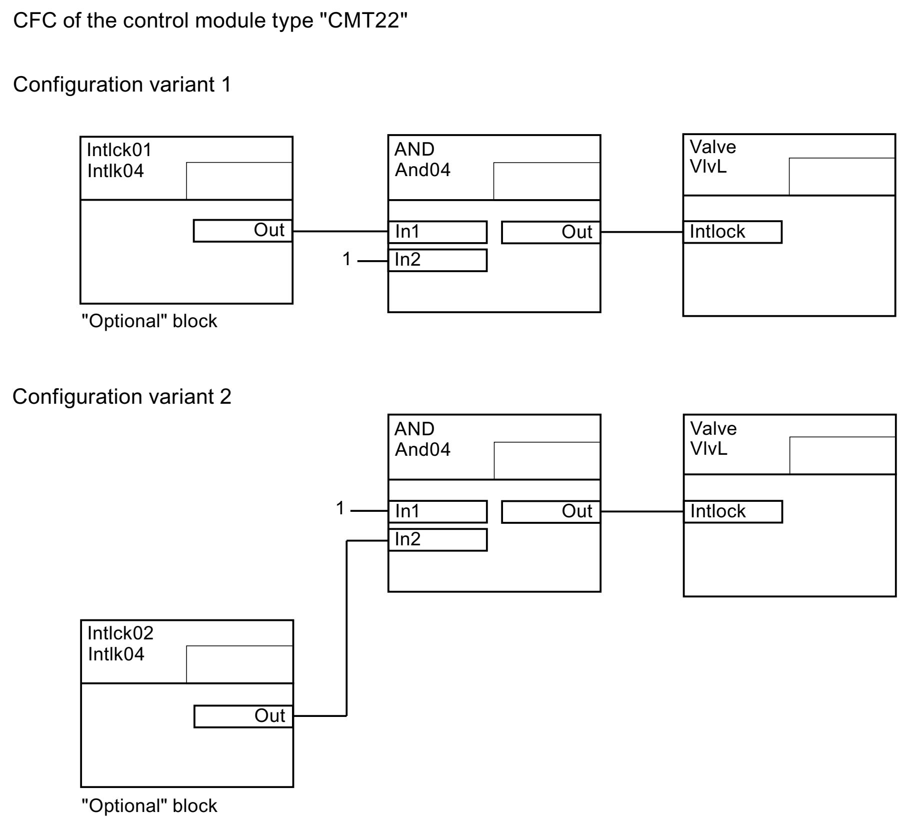
Configuration example 2

In this configuration example, the optional blocks are configured as follows in the control module type:

- In configuration variant 1, the block "Intlck01" is defined as optional.
- In configuration variant 2, the block "Intlck02" is defined as optional.
- The blocks "Valve" and "AND" are not optional.

Variant selection at the instance:

- For the application variant "A" with one "Intlck" block, only configuration variant 1 is activated in the instance.
- For application variant "B" with multiple "Intlck" blocks, both configuration variants 1 and 2 are activated in the instance.



Properties of this configuration example

Configuration:

- With this configuration method, a double interconnection target under the "Interconnection to" attribute – as in configuration example 1 at the "Intlck" parameter of the "Valve" block – is not required at any parameter.
- The "AND" block is not optional and is always present.
- The "Out" outputs of the optional "Intlck" blocks are assigned to different inputs of the "AND" block.

Layout of the structure in the technological editor:

The following table shows the associated structure in the technological editor. A sub-control module "Option_x" is created for each of the configuration variants 1 and 2. The inputs and outputs of the optional blocks that are to be interconnected are subordinated as parameters.

Symbol	Name/ symbol	Name/ symbol	Parameter name	Remark
	"CMT22"			Control module for configuration example 2
		"Option_1"		Sub-control module for parameters of configuration variant 1
			"Intlck01_Out"	Parameter for "Intlck01" block, "Out" output
		"Option_2"		Sub-control module for parameters of configuration variant 2
			"Intlck02_Out"	Parameter for "Intlck02" block, "Out" output
		"InputSelection"		Sub-control module for the "AND" block as interconnection target
			"And_In1"	Parameter for "AND" block, "In1" input
			"And_In2"	Parameter for "AND" block, "In2" input

Effects on the variant selection at the instance:

If the optional blocks for the application variants are activated at an instance of this control module type "CMT22", the chronological order of the activation does not play a role.

Because the "Out" outputs of the optional "Intlck" blocks are assigned to different inputs of the "AND" block, conflicts in the automatic interconnection do not occur.

11.1.16 Configuration options of the control module in the equipment module / equipment phase

Overview

A control module can be assigned to an equipment module/phase.

Configuration options

The following configuration options are available for assigning the control module:

- The control module to be assigned is integrated as a component of the equipment module/phase type and is therefore available as a lower-level object.
  If this control module is available as a type in the master data library, a "Control module assignment" is configured for this lower-level object "Control module". The so-called "role" is also defined in the "Control module assignment".
  If this control module is not available as a type in the master data library, it can be integrated as a lower-level object in the equipment module/phase even without an associated "Control module assignment". This type of control module makes sense, for example, for interlocks or calculations.

  Note

  A control module that is assigned a "role" via the "Control module assignment" cannot be designated as "optional".

  You can find additional information on this in the section "Integrating and assigning a control module in the type (Page 260)".
- The control module is not a component of the equipment module/phase.
  The "Control module (basic requirement)" and "Control module assignment" objects are then configured for this control module.
  - The minimum requirements for the control module to be assigned, for example, the required commands or feedback about the status, are described in abstract in the "Control module (basic requirement)".
    If the control module is configured at the type as a "Control module (basic requirement)", a specific control module from the project must be assigned at the instance. You can find additional information on this in the section "Assigning the control module instance to an equipment module/phase (Page 262)".
  - The so-called "role" is also defined in the "Control module assignment".

  You can find additional information on this in the section "Configuring and assigning a control module as a basic requirement (Page 258)".

Optional control modules

A control module can be designated as optional at the type of an equipment module/phase.

This means that this control module may also be omitted at an instance of the equipment module/phase in accordance with the real plant.

Note

When an optional control module that is defined as a process tag (e.g. calculations, interlocks, etc.) is omitted, the user must ensure that certain control strategies are prohibited at the instance.

Commands and status

By using commands and statuses, a control module can be addressed in the sequencers of the equipment module/phase.

You can find additional information on this in the section "Configuring a command or status at the type of the control module (Page 212)".

Converting to "basic requirement"

A control module that is a component of the equipment module/phase type can be defined as a "basic requirement". The "Basic requirement" attribute is set at the control strategy for this. Thereafter, the control module is no longer a component of the equipment module/phase type. With this procedure, the properties of the control module can be implemented in a basic requirement. In addition to the basic requirement, a "Control module assignment" is configured as described in the section above on configuration options.

See also

Basics of the equipment phase (Page 299)

Basics of equipment modules (Page 264)

Overview for configuring and managing the equipment phase (Page 311)

Overview for configuring and managing the equipment module (Page 280)

11.1.17 Configuring and assigning a control module as a basic requirement

Introduction

The assignment of a control module to an equipment module or equipment phase is configured in the technological editor of the equipment module/phase.

The configuration depends on whether the control module to be assigned is integrated in the equipment module/phase type as a component.

The following configurations are possible:

- "Configuration 1":
  The control module to be assigned is integrated as a component of the equipment module/phase type.
  The following objects are configured for this:
  - A control module as an object in the equipment module/phase, represented by the following icon:
    
  - A "control module assignment" object, represented by the following icon:
    

    You can find additional information on this in the section "Integrating and assigning a control module in the type (Page 260)".
- "Configuration 2":
  The control module to be assigned is not a component of the equipment module/phase type.
  The following objects are configured for this:
  - A corresponding "control module (basic requirement)" object, represented by the following icon:
    

    The minimum requirements for the control module to be assigned are abstractly described in the basic requirements.
  - A "control module assignment" object, represented by the following icon:
    

    The corresponding control module or a basic requirement and "role" are assigned and configured in the "control module assignment".

The following section describes the procedures for "Configuration 2", in which the control module to be assigned is not a component of the type.

You can find additional information about this in the section "Configuration options of the control module in the equipment module / equipment phase (Page 256)".

Requirement

An equipment module / equipment phase type is created in the master data library.

Procedure

1. Open the desired equipment module/phase type in the CFC editor.
2. Select the icon for an equipment module type, for example, in the technological editor.
   
3. Select "Insert new object > Control module (basic requirement)" from the shortcut menu.
   The control module requirement is inserted and displayed as an icon.
   

4. Select the icon for the control module requirement in the configuration editor.
   The associated attributes are displayed.
5. Configure the attributes, for example, the name and function identifier.
6. Select the icon for the equipment module type in the technological editor.
7. Select "Insert new object > Control module assignment" from the shortcut menu.
   The control module assignment is inserted and displayed as an icon with the following additional information.
   
8. Select the icon for the newly created control module assignment in the configuration editor.
   The associated attributes such as "Control module assignment" and "Role" are displayed.
   A basic requirement is configured for the "Control module assignment" attribute and no specific control module because this associated control module is not integrated as a component of the type.
   For configuring the "Assigned control module" attribute, drag the basic requirement created above to the "Control module assignment" object in the technological editor.
   The newly created assignment is displayed in the "Control module assignment" attribute.
9. Configure the "role" for the attribute with the same name.
   The "role" indicates how the associated control module or basic requirement is to be used in the context of this type.
   Enter a comment, if needed.
10. If you want to configure lower-level elements under the "Control module (basic requirement)", for example, the "Status", select the icon for the "Control module (basic requirement)" in the technological editor.
    
11. Select the "Insert new object" command from the shortcut menu.
    All available elements are shown in the shortcut menu.
    Select the appropriate menu command to insert the desired object, for example "Command".
    Repeat this step as often as necessary to add more lower-level elements.
12. Configure the attributes of the lower-level elements.

Result

The "Basic requirement" and the "Control module assignment" are configured at the equipment module/phase type.

In this case, the control module is not a component of the equipment module/phase type; it is abstractly described in the basic requirement.

11.1.18 Integrating and assigning a control module in the type

Introduction

The assignment of a control module to an equipment module or equipment phase is configured in the technological editor of the equipment module/phase.

The configuration depends on whether the control module to be assigned is integrated in the equipment module/phase type as a component.

The following configurations are possible:

- "Configuration 1":
  The control module to be assigned is integrated as a component of the equipment module/phase type.
  The following objects are configured for this:
  - A control module as an integrated object in the equipment module/phase, represented by the following icon:
    
  - A "Control Module Assignment" object, represented by the following icon:
    

    The corresponding control module and "role" are assigned and configured in the Control Module Assignment.
- "Configuration 2":
  The control module to be assigned is not a component of the equipment module/phase type.
  The following objects are configured for this:
  - A "Control Module Assignment" object, represented by the following icon:
    
  - A corresponding "Control Module (Basic Requirement)" object, represented by the following icon:
    

    You can find additional information on this in the section "Configuring and assigning a control module as a basic requirement (Page 258)".

The following section describes the procedures for "Configuration 1", in which the control module to be assigned is a component of the type.

You can find additional information about this in the section "Configuration options of the control module in the equipment module / equipment phase (Page 256)".

Requirements

The following are available in the master data library:

- An equipment module/phase type
- A control module type

Procedure

1. In the master data library, navigate to the desired equipment module/phase type and open it in the CFC editor.
2. To configure the associated control module, navigate to the desired control module type in the template catalog of the master data library.

3. Drag the desired control module type and drop it onto the equipment module/phase type in the technological editor.
4. The control module is inserted and displayed as an icon.
   
5. Select the icon for the equipment module type, for example, in the technological editor.
   
6. Select "Insert New Object > Control Module Assignment" from the shortcut menu.
   The Control Module Assignment is inserted and displayed as an icon with the following additional information.

	Name of the assigned control module	"(" <Role> ")"

7. Select the icon for the newly created "Control Module Assignment".
   The associated attributes such as "Control Module Assignment" and "Role" are displayed.
8. For configuring the "Control Module Assignment" attribute, drag the control module created above to the "Control Module Assignment" object in the technological editor.
   The newly created assignment is displayed in the "Control Module Assignment" attribute.
9. Configure the "role" for the attribute with the same name.
   The "role" indicates how the associated control module or basic requirement is to be used in the context of this type.
   Enter a comment, if needed.

Result

A control module is integrated into the equipment module/phase type and the "Control module assignment" is configured.

This control module is now a component of the equipment module/phase type.

11.1.19 Assigning the control module instance to an equipment module/phase

Introduction

A control module can be configured at the equipment module/equipment phase type as a component of the type or as a "Control module (basic requirement)".

1. If a control module is integrated as a component of the type, it is included in the copying when the instance is created.
2. If the control module is configured at the type as a "Control module (basic requirement)", a specific control module from the project must be assigned at the instance.

The following section describes the procedures for this option "2".

Requirements

- The master data library contains the desired type of the equipment module/phase.
- A control module as a "Control module (basic requirement)" and the corresponding "Control module assignment" are configured at this type.
- An instance is created from this type.

Procedure

1. In the CFC editor, open the desired instance that meets the requirements listed above.
2. Open the control module to be assigned in the CFC editor.
3. To get a better overview, arrange the two open windows so that both are visible.
4. Select the icon of control module in the window of the control module in the technological editor.

   
5. Drag this icon from the window of the instance and drop it onto the icon of the desired control module assignment in the technological editor.

   
6. The newly configured assignment appears in the attribute list of the "Control module assignment" attribute.
   At the same time, the interconnections between the instance and the associated control module are automatically created.
   Use the shortcut menu of this attribute to delete the assignment or to jump to the control module.

   Note

   If you delete the entry in the "Control module assignment" attribute, all the associated interconnections are deleted as well.

Result

A specific control module is assigned at the instance of an equipment module/phase.

The interconnections between the instance and the associated control module are created.

See also

Overview for configuring and managing the equipment phase (Page 311)

Overview for configuring and managing the equipment module (Page 280)

Configuration options of the control module in the equipment module / equipment phase (Page 256)

11.2 Configuring and using equipment modules

11.2.1 Basics of equipment modules

Introduction

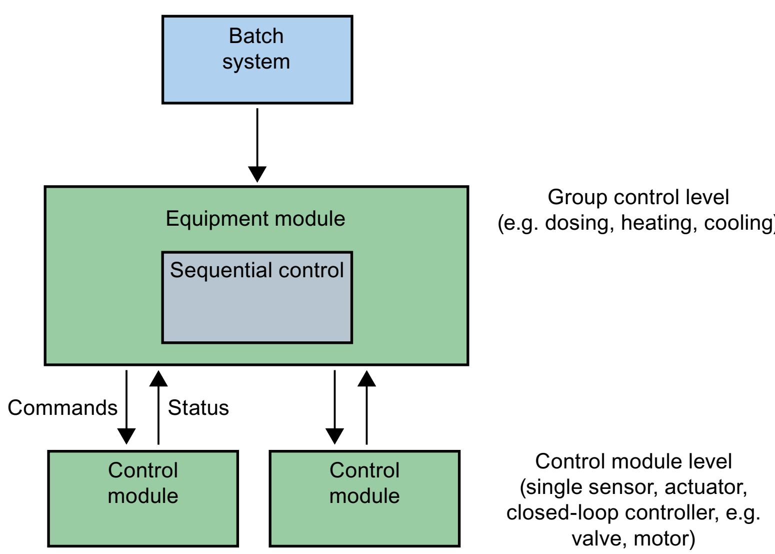
Terms

The English terms "Equipment Module (EM)" and "Equipment Phase (EPH)" are used in the international language environment according to the ISA-88 standard. In German-speaking countries, the terms "Technische Einrichtung" ("Equipment module") and "Technische Funktion" ("Equipment phase") are used. You can find more information about classification in the "ISA-88" standard in the section "Classification of the equipment module in the ISA-88 standard (Page 278)".

Concept of the equipment module

The "Sequential control" is also a component of an equipment module. Control strategies, setpoints, other parameters and the actual procedural control with the sequencers is defined in the "Sequential control". The actions and transition conditions of the sequencer can be described using commands and status.

A control module is used to assign the equipment module to the sensors and actuators on the control-loop level. Commands and states can be defined specifically at the control modules or globally in the master data library for the step and transition programming.



The configured mapping on the technological level in the system description between the equipment module and the control modules is automatically implemented in PCS 7 for the automation level in the corresponding CFC interconnections between the given blocks.

Although control modules can also be used for continuous processes, they are preferably used together with the equipment modules for batch processes.

Basics of the equipment module

Brief overview

The following description gives a brief overview of the data objects for equipment modules and control modules and the global command or status.

You can find more information on this in the section "Overview of data objects of the equipment module and control module (Page 269)".

Elements of the equipment module

- The "Sequential control" with its sequencers and parameter descriptions always has to be available in the equipment module, but only once.

The following objects can be optionally contained:

- Control module as an integral part of the type, several are possible
- Control module assignment; several are possible
- Control module requirements as basic requirements in which the minimum requirements for the control module are defined.
  Cannot be marked as optional, several are possible
- "Equipment module assignment"; several are possible
- "Equipment module (basic requirement)"; as basic requirement in which the minimum requirements for an equipment module are defined; several are possible
- The following objects are possible as control variables, several are possible:
  - Parameters
  - Signals
- Messages; several are possible

Elements of the control module

The control module can optionally contain the following objects:

- Sub control modules; several are possible
- Commands; several are possible
- Statuses; several are possible
- The following objects are possible as control variables, several are possible:
  - Parameters
  - Signals
- Messages; several are possible

Commands and status

The "Command" and "Status" objects can be defined as follows:

- At the type of a control module (not at an instance) or at a "Control module (basic requirement)".
  These commands and statuses are specific to these objects.
- Globally, i.e. with the SIMATIC Manager in the master data library.
  These global commands and states are generally based on equipment phases and can be used in all equipment phases of a plant.

You can find more information on this in the section "Overview of data objects of the equipment module and control module (Page 269)" in the tables for the data objects of the control module and the globally defined data objects.

The control module in the equipment module

A control module can be assigned to an equipment module.

You can find additional information on this in the section "Configuration options of the control module in the equipment module / equipment phase (Page 256)".

Type and instance of an equipment module

The type/instance concept is used for repeated usage. The advantage of the type/instance concept is the ability to make changes to the type at a central location and then transfer the changes by synchronizing the instances.

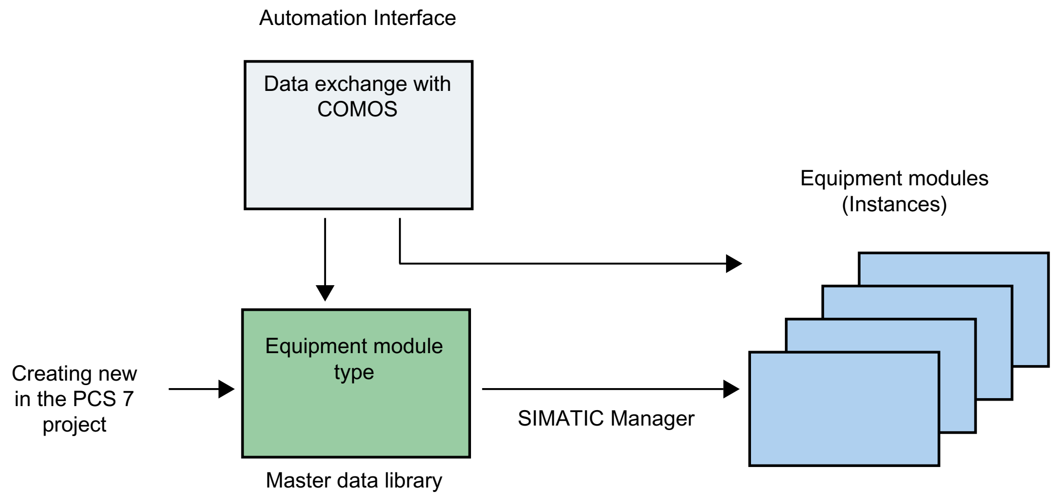
- Synchronization is available when there are differences between the type and instance. Instance-specific extensions are managed as such and are not lost when the type and instance are synchronized. Changes can therefore be bumplessly loaded in the automation system.
- The type and instance of equipment modules and control modules are included in the data exchange with COMOS Integrated Engineering.

Type and instance:

- A type of an equipment module can only be created in the master data library of the PCS 7 project using the following options:
  - In the "Plant view" in the SIMATIC Manager
  - By creating it in COMOS Integrated Engineering and subsequently importing it via the automation interface (AI).

  In the master data library, the type of equipment module is configured for use in the PCS 7 project. The type can be copied and deleted in the master data library using the SIMATIC Manager .
- The instances are the repeated uses of a type in the plant description of the plant hierarchy. Instances of a type are created simply by copying the type from the master data library into the project under a node in the plant hierarchy.
  When the control modules are assigned to the instances, the connections to the control-loop level and the specific control modules are created as well.

The following figure shows the ways to create a type and instances.



Naming the type and instance

- The name of the type is unique within the master data library, exactly like the control module types.
- The types of equipment modules share a namespace in the master data library with the following objects:
  - Type of a control module
  - Global commands and status

  Therefore, an equipment module type cannot have the same name as a control module type.
- Name of a equipment module type:
  The maximum permitted length of the name is 22 characters.
  It is recommended to use only 18 characters to provide some reserve for automatic renaming, for example, by the automatic addition of the character string "(1)" to avoid name collisions during copying.
- The instances of the equipment module type are hierarchically addressed through the plant hierarchy. Higher-level units are therefore unique.

Note

If the name already exists during copying:

If you copy a chart and paste it, the chart name suffixed with "(x)" (where x is 1, 2, 3, 4, and so on) is assigned as the name of the copied chart.

For example:

If there exists a chart named "ChartName" and if you create a copy of this chart, the copied chart is named as "ChartName(1)" because the name "ChartName" already exists. Further, if you paste it again (create another copy), the next copied chart is named as "ChartName(2)". But if you create a copy from the "ChartName(2)" chart then the copied chart will be named as "ChartName(2)(1)".

Note

If the maximum length of the name is exceeded during copying:

If you copy a chart which is named with 22 characters (maximum length) and paste it, the name of the copied chart is shortened from the end of the string until "~(x)" can be appended (where x is 1, 2, 3, 4, and so on).

For example:

If there exists a chart named "ChartNameMotorControlA" (22 characters) and if you create a copy of this chart then the copied chart is named as "ChartNameMotorCont~(1)".

By the general rules of naming in the case where the name already exists, the new chart name should have been "ChartNameMotorControlA(1)" (25 characters) but this exceeds the maximum permitted length of 22 characters. Therefore, the name is shortened to 22 characters with the methodology explained above.

Comparison and synchronization of type and instance

You can information on this in the section "Comparing and synchronizing the type and instances of an equipment module (Page 286)".

Data exchange with COMOS

The import functions of the automation interface (AI) can be used to transfer the types of equipment modules (basic functions) defined on the engineering level with COMOS Integrated Engineering to the PCS 7 process control system. The associated sequencers, transition conditions and actions are imported along with the types.

This enables the data generated on the engineering level in COMOS in the form of equipment modules to be imported into PCS 7 and thereby entered in the automation layer. There, the data appear as sequencers (SFC) and automation programs (CFC).

You can find additional information about this in the section "Overview of data exchange with COMOS (Page 332)".

11.2.2 Overview of data objects of the equipment module and control module

Introduction

The following tables describe:

1. Data objects of the "Equipment module"
   Icon in the technological editor:
   
2. Data objects of the "Sequential control"
   Icon in the technological editor:
   
3. Data objects of the "Control module"
   Icon in the technological editor:
   

Data objects of the equipment module

An equipment module consists of the following elements, of which there can be one or several.

Note

No "Command" and "Status" objects can be defined in an equipment module. They can be defined either specifically for a control module type or basic requirement, or as "global" objects. You can find more information on this in the tables below.

Object/icon	Description
Sequential control 	The sequential control defines the sequencers and their parameter descriptions (for example, setpoints, control strategies, parameters, messages). The sequential control may only be in the equipment module once. Each equipment module must be assigned one and only one sequential control. The elements are described in the table below, "Data objects of the sequential control".
Control module 	A control module can be configured as component of an equipment module type. Several control modules can be present. Optional control modules: • Control modules can only be defined as "optional" at the type in the master data library. • Control modules can only be defined as "optional" on the first sub-level under an equipment module type. • The nesting depth for control modules under the equipment module is limited to a maximum of 3 levels. This means that an equipment module can only have a maximum of two sub-levels of sub-control modules, for example, the levels "Equipment module > Control modules > Sub-control modules". **Note:** A control module that is a component of the equipment module type can be defined as a "basic requirement". The "Basic requirement" attribute is set at the control strategy for this. The control module is then no longer a component of the equipment module type and the blocks of the control module are deleted from the model.
Control module assignment 	If a control module is to be controlled in the sequencers, the equipment module must be assigned to the control module. The assignment configures the following: 1. The "role". The so-called "role" of the control module is set in the equipment module type, but no specific control module. The "role" indicates how the assigned control module is used in the context of this type, e.g. that an assigned "valve" control module has the "role" of an "outlet". The "role" is used to address the target control module in the actions and transition conditions of the sequencers. 2. The link to a control module or basic requirement. – When the control module is integrated in the type of the equipment module, this control module is assigned. – If the control module is not integrated in the type, a corresponding basic requirement is assigned. The specific requirements for the control module to be assigned are specified themselves under "Control module (basic requirement)" described below. The assignment can be provided with a comment. There can be more than one assignment. If a "Control Module Assignment" is configured, you must also create a corresponding basic requirement or a control module that is integrated in the type.

Object/icon	Description
Equipment module assignment 	The assignment configures the following: 1. The "role". The type of the equipment module defines the "role" of an equipment module, but no specific equipment module. The "role" indicates how the assigned equipment module is used in the context of this type, for example, that an assigned "valve" equipment module has the "role" of an "outlet". The "role" is used to address the target equipment module in the actions and transition conditions of the sequencers. 2. The link to an equipment module or basic requirement. – If the linked equipment module is configured in the type of equipment module, this equipment module is assigned. – If the linked equipment module is not configured in the type, a corresponding basic requirement is assigned. The specific requirements for the equipment module to be assigned are specified themselves under "Equipment module (basic requirement)" described below. The assignment can be provided with a comment. There can be more than one assignment. If an "Equipment module assignment" is configured, you must also create a corresponding basic requirement or a lower-level equipment module that is integrated in the type.
Control module (basic requirement) 	The minimum requirements for the control module to be assigned are abstractly described in the "Control module (basic requirement)". The requirements include, in particular, the necessary commands, but also feedback about the status. • There can be more than one basic requirement. • The basic requirements are needed at the equipment module type to enable the use of commands and states, for example, in the sequential logic. • The "Command" and "Status" objects can be defined at the basic requirement. You can find additional information on this in the section "Special considerations for the implementation of specific commands/statuses (Page 213)". • A basic requirement cannot be defined as optional.
Equipment module (basic requirement) 	The minimum requirements for the equipment module to be assigned are abstractly described in the "Equipment module (basic requirement)". The requirements include, in particular, the necessary commands, but also feedback about the status. • There can be more than one basic requirement. • The basic requirements are needed at the equipment module type to enable the use of commands and states, for example, in the sequential logic. • The "Command" and "Status" objects can be defined at the basic requirement. You can find additional information on this in the section "Special considerations for the implementation of specific commands/statuses (Page 213)". • A basic requirement cannot be defined as optional.

Object/icon	Description
Parameters	• There can be several parameters. • The "Parameter" object can be used: – As a sub-object of an equipment module or control module. – As a formal parameter of commands and statuses. In the "Sequential control", these formal parameters define placeholders for specific parameters or constants at the points of use in the actions and transitions of the sequencer. • The "Value" and "Signal" attributes and the "Interconnection" relation can only be entered as alternatives. • The "Signal" attribute is only allowed for input parameters and is disabled for output parameters. • The "Negation" attribute is only allowed for input parameters of the "BOOL" data type and is disabled for output parameters. For input parameters, the "Negation" attribute relates to the parameter interconnection or the signal.
Signals	• There can be several signals. • The attributes of signals are represented in the same way as parameters in the technological editor. The input fields of the "Value", "Enumeration", "Operation identifier", "Unit" and "Interconnection" values or relations are disabled, however. • The "Negation" attribute is only allowed for input parameters of the "BOOL" data type and is disabled for output parameters. It applies to the "Signal" attribute for signals.
Messages	An equipment module can generate messages by default. • There can be several messages. • Messages can be configured directly under the equipment module. They then apply only to the block instance of the SFC type in the associated CFC. To assign the corresponding message, the message inputs of this block instance of the SFC type must be set as visible.

Data objects of the sequential control

The "Sequential control" consists of the following elements, of which there can be one or several.

Object/icon	Description
Parameters Used for the parameter description	• The parameter descriptions of the "Sequential control" correspond to the "characteristics", which are set at the SFC type. • In the context of integration on the engineering level (COMOS Integrated Engineering), there is a limitation to the parameters and attributes required for the process engineering in the data exchange models. • Parameters can be used as sub-objects of a "Sequence control" to define as individual variables the "Sequence control" interface externally and internally to the sequencers. You can find more information on this at the end of this table.
• Control strategies	Various process engineering procedures can be defined using control strategies. The control of sequencers is basically a function of the control strategies. Control strategies are important for higher-level recipe control level (batch systems). Attributes: • Name • Number • Comment • Assigned setpoint The assignment of a setpoint to a control strategy is performed in the COMOS data exchange model using a relation (assigned control strategy).
• Setpoints "Setpoints")	Setpoints can be specified in process engineering by manual operation or by a higher control level (batch system). Setpoints are assigned to individual control strategies. The actual (process) value is always offered as a control variable in addition to the setpoint. Attributes: • Name • Data type • Comment • Low limit • High limit • Unit High, low limit and unit (of measure) are typically determined or adapted at the instance of the equipment module.
• Process values / actual values	The integration of process values in the equipment module is used to control sequencers, e.g., using an actual value for step-enabling the sequencers. Attributes: • Name • Data type • Comment • Unit The unit is typically set at the instance of the equipment module.

Object/icon	Description
• Control values	Control values are required to connect external control modules, which are not directly accessible through the control module contacts. Attributes: • Name • Data type • Comment • Unit The unit is typically set at the instance of the equipment module.
• Parameters	A parameter is used to influence the behavior of a "Sequential control" in an instance, for example, with options. These parameters can also be used to set limits for specific instances. Attributes: • Name • Data type • Comment • Unit
• Bit memory	Bit memory is needed to temporarily store values. It is only used locally in the sequential control system. Attributes: • Name • Data type • Comment
• Timers	Timers are used, for example, for time monitoring or calculation of run time. Attributes: • Name • Comment
• Note texts	Note texts are used, for example, for the user interface. Attributes: • Name • Text

Object/icon	Description
• Position texts	Position texts are used to display the current procedural status on the operator station. Attributes: • Name • Text
Sequencer	The process engineering task is defined in a sequential logic. This is described using sequencers. The behavior must be defined for each control strategy for each state of the "Sequential control". The sequencers have exactly one start step and one end step. Sequencers can include the following elements, which are familiar from the SFC type: • Steps • Transitions • Alternative branches • Simultaneous branches • Loops • Jumps Both alternative and simultaneous branches are symmetrically merged again. The actions that are defined for a step on the process engineering level in COMOS are described with commands in the form of a list. The commands used for this are provided centrally on the control module, for example, motor, valve. Commands can be defined as well at the equipment module. The transition conditions are set on the process engineering level in COMOS as expressions with a Boolean result using the status and logical standard functions. For this, the status (or states) provided by the control modules, for example, motor, valve, can be used as well as the status defined at the equipment module.

Parameters as individual variables for step and transition configuration

In a sequencer, various individual variables, for example, actual values, limit values or similar, may be required for step and transition configuration.

To create these individual variables, the required block I/O of the block instance of the corresponding SFC type is selected and dragged to the "Sequential control" in the technological editor. A "Parameter" object is automatically created. The names of these individual variables are pre-defined and cannot be changed.

These individual variables are interconnected and assigned parameters at the instance of the equipment module.

Data objects of the control module

One or more control modules can be assigned to an equipment module.

In connection with the equipment module, it is particularly important that control modules be accessible via commands and states of the equipment module sequencers.

The control module consists of the following elements, which may or not be present:

Object/icon	Description
Commands	• A command can only be defined as an object of a control module at the type of the control module (not at an instance), and is therefore type-specific. You can find additional information on this in the section "Special considerations for the implementation of specific commands/statuses (Page 213)". • A command can be provided with one or more formal parameters. • A control module command is determined by its name. • Each command has the attributes "Comment" and "Author". • There can be several commands. Example of a command: A command named "Open" is defined for opening a valve. Commands can also be provided with parameters, e.g., if certain information (values, variables) are first known at the instance. An example of a command with parameters is the specification of motor speed: • "Set_Motor_Speed(Speed)"
Status	• A status can only be defined as an object of a control module at the type of the control module (not at an instance), and is therefore type-specific. You can find additional information on this in the section "Special considerations for the implementation of specific commands/statuses (Page 213)". • A status can be provided with one or more formal parameters. • The status of a control module command is determined by its name. • Each status has the attributes "Comment" and "Author". • There can be several states. Example of a status: A status named "Closed" is created for the closed state of a valve.
Parameters	• There can be several parameters. • The "Parameter" object can be used: – As a sub-object of an equipment module or control module. – As a formal parameter of commands and statuses. In the "Sequential control", these formal parameters define placeholders for specific parameters or constants at the points of use in the actions and transitions of the sequencer. • The "Value" and "Signal" attributes and the "Interconnection" relation can only be entered as alternatives. • The "Signal" attribute is only allowed for input parameters and is disabled for output parameters. • The "Negation" attribute is only allowed for input parameters of the "BOOL" data type and is disabled for output parameters. For input parameters, the "Negation" attribute relates to the parameter interconnection or the signal.

Object/icon	Description
Signals	• There can be several signals. • A signal defines an I/O channel request. • The attributes of signals are represented in the same way as parameters in the technological editor. The input fields of the "Value", "Enumeration", "Operation identifier", "Unit" and "Interconnection" values or relations are disabled, however. • The "Negation" attribute is only allowed for input parameters of the "BOOL" data type and is disabled for output parameters. It applies to the "Signal" attribute for signals. • The attributes “High scale value”, “Low scale value”, and “Unit” are integral part of an analog signal. • Scale and unit attributes are implicitly linked by the system, but can also be explicitly assigned to the corresponding block variables from the channel driver blocks if required. • The corresponding scale with respect to the unit parameters from the channel driver blocks are not required to be defined as independent CM parameters.
Sub-control modules	• A control module can be subordinate to other control modules. • There can be several sub-control modules. • The nesting depth for control modules is limited to a maximum of 2 levels. This means that a control module can only have one sub-level of sub-control modules. • Sub-control modules can only be defined as "optional" on the first sub-level under a control module/equipment module type. You can find more information on this in the above table of the data objects of an equipment module.
Messages	• An equipment module can generate messages by default. • There can be several messages. • Read the following information for configuring messages.

Configuration note for messages of a control module or sub-control module

A message and the associated reporting block should always be configured in the same control module or control module type. This also applies to sub-control modules.

If the message is created directly in a control module type, but the associated reporting block is contained in a sub-control module of this type, after an export of the instances of this control module type from COMOS to PCS 7 the message will not be assigned to the reporting block.

The lack of assignment can be simply solved using the "Plant types > Synchronize" function in the shortcut menu of the project. However, to avoid the problem of this missing assignment after export, this configuration note should be taken into account.

See also

Basics of equipment modules (Page 264)

Basic information on control modules and their types (Page 203)

Configuring a global command or status (Page 348)

Overview of the data objects of global "Command" and "Status" (Page 352)

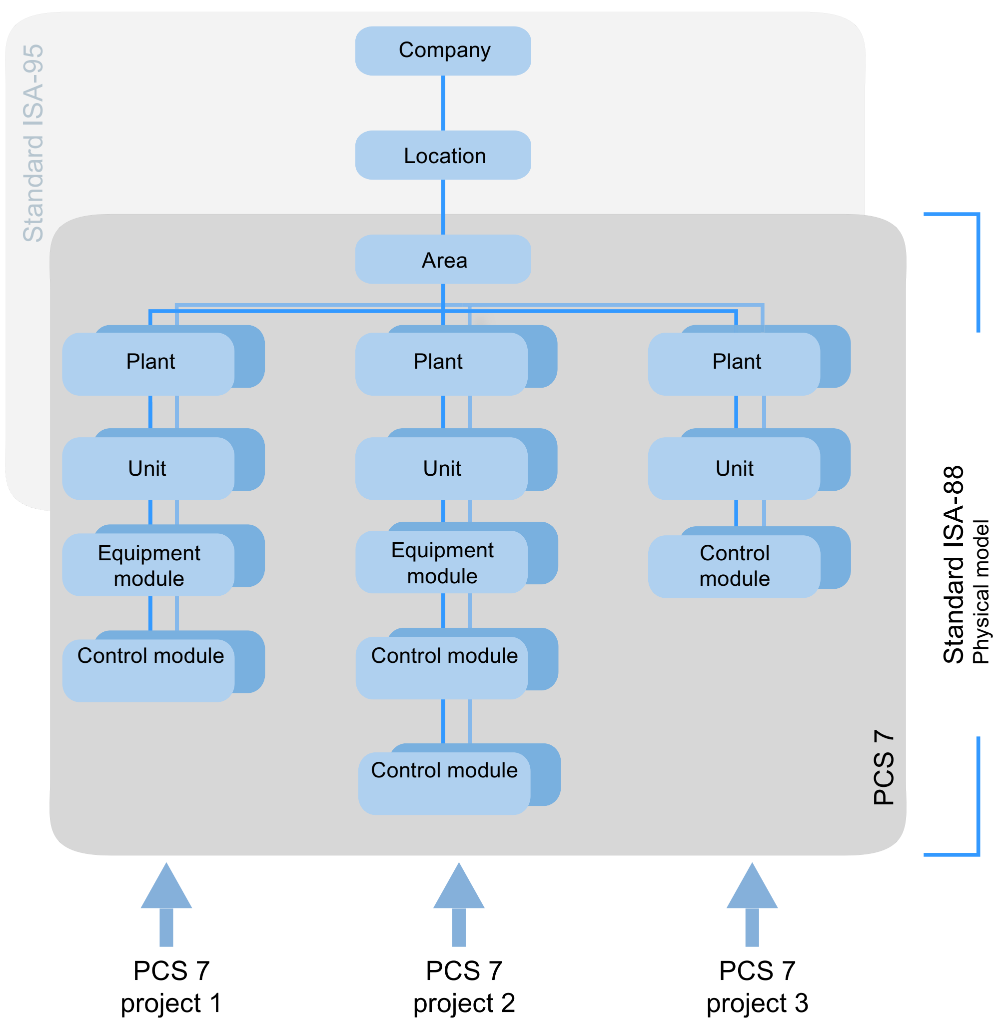
11.2.3 Classification of the equipment module in the ISA-88 standard

Overview

An equipment module is classified in the physical plant model of the ISA-88 standard (IEC 61512-1) below a unit.

- When using SIMATIC BATCH, a node must be labeled in the plant hierarchy as an equipment module according to ISA-88. This is why the possibility provided by the existing standard to classify an equipment module directly under a process cell is not supported in the context of PCS 7.
- Control modules can be either below an equipment module or on the level directly below the unit.

The following figure shows the placement of the equipment module in the so-called physical model of the ISA-88 standard. Examples of three possible configurations for a PCS 7 project are presented.



Terms of the ISA-88 standard

Process cell

A plant or process cell contains all of the equipment required to make a batch. Process cells are often divided into lines. A line consists of all units and other devices that can be used by a specific batch. Lines can be left unchanged from batch to batch, or be defined differently for each batch.

In a PCS 7 multiproject, only one "Plant" hierarchy folder can be created. Examples for this include polymerization plants, dying works, multi-purpose plants.

Unit

A unit is an independent group of equipment modules that are usually arranged around essential process equipment, for example, a mixing tank or a reactor.

A unit has the following characteristics:

- A unit can perform one or more key process activities, for example, reactions, crystallization and preparation of solutions.
- Units operate relatively independent of one another.
- A unit frequently contains a complete material batch at some point in the process flow of the batch.
- A unit cannot process more than one batch at a given time.

Equipment module

An equipment module can consist of control modules and lower-level equipment modules. An equipment module is usually located in a process device, for example, a filter.

Characteristics of an equipment module:

- Can belong to a unit or incorporate an independent group of equipment in a process cell.
- Can perform a finite number of specific, minor processing functions, for example, dosing or weighing.
- Can optionally contain the raw materials for a batch.

Control module

A control module is usually a collection of sensors, actuators and other control modules and associated process equipment that can be operated as an individual device from the perspective of the control engineering.

A control module can also be composed of other control modules. For example, a dosing control module can be defined as a combination of several automatic switching valve control modules.

See also

Basics of equipment modules (Page 264)

11.2.4 Creating, configuring and managing equipment modules

11.2.4.1 Overview for configuring and managing the equipment module

Overview

- Configuring
  A type of an equipment module can only be created in the master data library of the PCS 7 project using the following options:
  - In the "Plant view" in the SIMATIC Manager
  - By creating it in COMOS Integrated Engineering and subsequently importing it via the automation interface (AI)

  In the master data library, the type of equipment module is configured for use in the PCS 7 project.
  You can find additional information on this in the section "Configuring an equipment module (type) (Page 281)".
- Management
  An equipment module type can be copied and deleted in the SIMATIC Manager.
- Configuring the sequential control of the equipment module
  You can find additional information about this in the section "Configuring the sequential control of an equipment module (Page 283)".
- Control modules on the type of equipment module
  Control modules can be assigned as a lower-level objects in the equipment module, or integrated into the equipment module.
  You can find additional information on this in the section "Configuration options of the control module in the equipment module / equipment phase (Page 256)".
- Creating an instance
  An instance can be created by copying the equipment module type from the master data library into a node of the plant hierarchy.
  You can find additional information on this in the section "Creating an instance of the equipment module type (Page 285)".
- Type/instance synchronization
  Synchronization of the type and the instances may be required when changes are made to the type or its associated instances.
  You can find additional information on this in the section "Comparing and synchronizing the type and instances of an equipment module (Page 286)".
- Control modules at the instance of the equipment module
  If the control module is configured at the type as a "Control module (basic requirement)", a specific control module from the project must be assigned at the instance.
  You can find additional information about this in the section "Assigning the control module instance to an equipment module/phase (Page 262)".

11.2.4.2 Configuring an equipment module (type)

Introduction

A type of an equipment module can only be created in the master data library of the PCS 7 project using the following options:

- In the "Plant view" in the SIMATIC Manager
- By creating it in COMOS Integrated Engineering and subsequently importing it via the automation interface (AI)
  You can find additional information about this in the section "Overview of data exchange with COMOS (Page 332)".

The type can be copied and deleted in the master data library using the SIMATIC Manager.

In the master data library, the type of equipment module is configured for use in the PCS 7 project.

This procedure is described below.

Requirements

- A multiproject containing a master data library is open in the SIMATIC Manager.

Procedure

1. Use the menu command "View > Plant View" to open the plant view.
2. Once the equipment module is imported, proceed to step 4.
3. If you wish to create a new equipment module type, in the plant view of the master data library navigate to the directory in which the object is to be created, for example, "Process tag types".
   Select the "Insert New Object > Equipment Module (Type)" command from the shortcut menu.
   Proceed to step 5.
4. In the plant view of the master data library, navigate to the directory where the equipment module was created or stored after import, for example, "Process tag types".
   You can move or copy the type into another directory you have created in the master data library and configure the properties there.
5. The equipment phase type appears in the right window with the corresponding icon.
   
6. Double-click on the icon. The CFC editor opens; this where the properties of the equipment module type are configured.
   Open the technological editor. The type has different attributes, such as "Name", "Author", "Comment".
   Enter the desired data for the attributes.
7. If you want to configure lower-level elements under this type, select the icon for the type of the equipment module in the technological editor.
   

8. Select the "Insert New Object" command from the shortcut menu.
   All available elements are shown in the shortcut menu.
   – "Control module"; for a sub-control module
   – "Control module (basic requirement)"; as basic requirements in which the minimum requirements for the control module are defined
   – "Control module assignment"
   – "Equipment module assignment"
   – "Equipment module (basic requirement)"; as basic requirement in which the minimum requirements for an equipment module are defined
   – "Parameter"
   – "Signal"
   – "Message"
   – "Sequential control";
     the sequential control may only be in the equipment module once. Therefore, the menu command is enabled only when no "Sequential control" object is currently present in this equipment module. You can find information about configuration in the section "Configuring the sequential control of an equipment module (Page 283)".
9. Select the appropriate menu command to insert the desired object, for example, "Parameter".
10. Repeat this step as often as necessary to add more lower-level elements.

Result

The equipment module type is configured.

Lower-level elements such as parameters and control modules are inserted.

Note

The objects, for example, the values of attributes or assignments still have to be configured in the technological editor.

See also

Configuring and assigning a control module as a basic requirement (Page 258)

Integrating and assigning a control module in the type (Page 260)

11.2.4.3 Configuring the sequential control of an equipment module

Introduction

The sequential control may only be in the equipment module once.

- When a new equipment module is created, a "Sequential control" object and an associated SFC type are created automatically. This assigned SFC type defines the sequencers and their parameter descriptions (for example, setpoints, control strategies, parameters, messages).
- If the "Sequential control" object is deleted in the equipment module and added again, an SFC type must once again be assigned to the object.
- The SFC type that is assigned to a "Sequence control" object cannot be deleted in the "Charts" directory of the master data library as long as the associated "Sequence control" object exists in an equipment module.

Requirements

- A multiproject containing a master data library is open in the SIMATIC Manager.
- An equipment module type is available in the master data library and configured as described in the section "Configuring an equipment module (type) (Page 281)".

Procedure

1. In the master data library, open the equipment module type in the CFC editor.
2. Select the icon for the equipment module type in the technological editor.

   All known objects are displayed under this icon.
   In a newly created equipment module, the "Sequential control" object and an associated SFC type are automatically available.
   If the "Sequential control" object already exists, check in the technological editor if there is an entry in the "Assigned" column of the "Assigned block" attribute. If there is an entry and an SFC type is therefore assigned, proceed to step 6.
3. If the "Sequential control" object is not available, select the "Insert New Object" command from the shortcut menu. All available objects are shown in the shortcut menu.
   Select the "Sequential control" menu command.
   The sequential control is inserted and displayed as an icon.

   Because each equipment module can be assigned one and only one sequential control, the corresponding menu command is disabled in the shortcut menu of the equipment module afterwards.
4. Select the icon for the "Sequential control" in the technological editor. The right window shows the associated attributes.
   Select the "Assigned block" attribute. When creating the sequential control manually, there is no entry in the "Assignment" column and thus no assigned SFC type.

5. To assign an SFC type, open the block catalog in the CFC editor.
Drag the desired SFC type from the block catalog and drop it in the chart window of the CFC editor. The newly inserted SFC type is displayed in the CFC and referenced in the "Assigned block" attribute.
If no SFC type appears in the block catalog, create the desired type in the "Charts" folder of the master data library in the component view.
6. If you want to configure lower-level elements for this sequential control, select the icon for the sequential control in the technological editor.
Select the "Insert New Object" command from the shortcut menu. All available objects are shown in the shortcut menu. Only "Parameter" can be selected in the current version.
7. Select the "Parameters" menu command in the shortcut menu.
The "Parameters of the equipment phase" dialog opens. All the parameters that can be used are shown in this dialog.
Select the required parameter and click "Insert". The parameter is inserted and displayed as an icon. Configure the attributes of this parameter.
8. Parameters can also be used as individual variables for configuring steps and transitions, e.g., for actual value, limit values.
If you would like to create a parameter as an individual variable, select the desired block I/O of the block instance of the corresponding SFC type and drag it to the sequential control in the technological editor.
The "Parameter" object is created automatically. The names of these individual variables are pre-defined and cannot be changed. You can find additional information in the section "Overview of data objects of the equipment module and control module (Page 269)" in the table of the data objects of a sequential control.

Result

The "Sequential control" object is configured at the type of an equipment module and an SFC type is assigned.

Parameters are created as subobjects of this sequential control, if required.

Note

For the assigned SFC type, you still need to configure the contents in the SFC editor, for example, the sequencer and characteristics.

11.2.4.4 The control module in the equipment module

Overview

A control module can be assigned to an equipment module.

The following configuration options are available:

- The control module to be assigned is integrated as a component of the equipment module type and is therefore available as a lower-level object.
  For this, the control module is configured as an object directly in the equipment module. A "Control module assignment" can also be configured in which the "Role" is defined.
- The control module is not a component of the equipment module type.
  The "Control module (basic requirement)" and "Control module assignment" objects are then configured for this control module.

You can find additional information about this in the section "Configuration options of the control module in the equipment module / equipment phase (Page 256)".

11.2.4.5 Creating an instance of the equipment module type

Overview

The equipment module type is stored in the master data library.

A type can be copied within the master data library or copied to create an instance in the project.

Properties when copying an equipment module type in the master data library

- When a type is copied within the master data library, a new type is created and all the components of the original type, including the basic requirements, are copied along with it.
- A control module assignment defined at the type is copied as well, regardless of whether the assigned control module therein is a component of type or a basic requirement.

Properties when creating an instance (copying from the master data library into the project)

- An instance can be created by copying the equipment module type from the master data library into a node of the plant hierarchy.
  This node can be classified as an equipment module according to the ISA-88 standard.
  If SIMATIC BATCH is used, however, this node must be labeled as an equipment module!

  Note

  Number of instances in a process cell node

  Any number of instances of an equipment module type can be created under a process cell node in the plant hierarchy. If this node is classified as an equipment module according to ISA-88, however, multiple instances do not make sense. However, this is not checked by the SIMATIC Manager and is thus the user's responsibility!

- Optional control modules are not copied.
  These can be selected and connected at the instance.
  Lower-level control modules can only be defined as "optional" on the first sub-level under the equipment module.
- A control module assignment defined at the type is copied as well, regardless of whether the assigned control module therein is a component of type or a basic requirement.
- A control module integrated as a component in the type is also copied when the instance is created.

- A "Control module (basic requirement)" is not included in the copying because the basic requirement only abstractly describes the minimum requirements for the control module to be assigned.
  The corresponding control modules must be assigned at the instance for these basic requirements.
- Commands and status can be defined at the types of equipment modules and control modules. These are not included in the copying.
- The corresponding type is known at the instance in order to support the type-instance synchronization.

Comparison between type and instance

Synchronization of the type and the instances may be required when changes are made to the type or the associated instances.

You can find additional information on this in the section "Comparing and synchronizing the type and instances of an equipment module (Page 286)".

11.2.4.6 Comparing and synchronizing the type and instances of an equipment module

Introduction

The type and instance of an equipment module are the same when the instance is created.

Synchronization of the type and the instances may be required when changes are made to the type or its associated instances.

Thus, for example, the sequencer of the equipment module defined in an SFC type is identical when the instance is created. If changes are made to the SFC type in the master data library, however, synchronization has to be performed as usual by updating the SFC type in the project.

Comparison and synchronization

During synchronization, all instances of a type in the project are compared to the corresponding type in the master data library. This shows the differences between the types of equipment modules and their instances (process engineering level) and between the associated objects, such as CFCs (program level).

The synchronization of parameter descriptions and sequencers between the master data library and project is accomplished by updating the SFC type in the project. You can find additional information about this in the section "How to update block/SFC types in the multiproject (Page 110)".

Note

As of CFC V9.0, a VXM license as of V9.0 is required for type instance synchronization of technological objects, e.g. control modules (CMT).

You can find additional information and notes on synchronization in the section "Notes on type instance synchronization of technological objects (Page 347)".

Included objects

The following lower-level objects are included in the synchronization of the type and instances of equipment modules on the plant level:

- Control modules
- Control variables:
  - Parameters
  - Signal
- Messages
- Commands and status
- Assignments of control modules
- Configuring a sequential control

Basic requirements for control modules, however, are generally not involved as they are by definition not permitted at the instances.

Note

Synchronization within the master data library

If a control module type is used in the library as a lower-level control module at the type of an equipment module, this lower-level control module is also synchronized with its control module type.

Procedure

1. Select the project in the plant view.
2. Open the shortcut menu and select the menu command "Plant types > Synchronize...".
   The "Synchronize plant types" dialog box opens.
3. Select the desired types of equipment modules or control modules as well that you want to compare and synchronize in the left column of the table.
4. Click the "Synchronize..." button to start the comparison.
   Only relevant attributes and relations are included in the comparison. You can find additional information on this in the section "Relevant attributes for the type/instance synchronization of equipment modules (Page 288)".
   Functions and function blocks that have been added to an instance are not included in the comparison and are suppressed in the presentation of the comparison result.

5. The comparison result is displayed.
   The following objects are highlighted in the comparison result:
   – Objects added at the instance, for example sub-functions, messages, functions and function blocks, are marked as additional objects.
   – Deleted objects.
   – Objects with modified attributes.

   Differences in the topology of the sequencers and parameter descriptions are not shown directly in the comparison result. Rather, the change time stamps of the two compared objects are displayed:
   – The change time stamp of the SFC type in the master data library that is associated with the "Sequential control".
   – The change time stamp of the corresponding SFC type in the project.

   For a control module that is integrated in the type of the equipment module, the corresponding "control module assignment" is also shown in the comparison result. If the "control module assignment" is missing or has been changed, you are prompted by an entry in the log file to check the control modules in the context of this equipment module.
6. In the comparison result, select the instances to be synchronized to the corresponding type. Select or clear the check boxes next to objects at the left in the dialog.
7. Start the synchronization with the "Synchronize templates" icon.
   The selected instances are synchronized.
   Instance-specific extensions are retained if they do not apply to objects of the type.

Result

A comparison of the types of equipment modules or control modules and their instances has been performed and selected instances have been synchronized.

See also

Basics of equipment modules (Page 264)

Creating an instance of the equipment module type (Page 285)

11.2.4.7 Relevant attributes for the type/instance synchronization of equipment modules

Introduction

A comparison or synchronization between the type and the corresponding instances may be required when changes are made to the type or the associated instances of an equipment module/control module.

The comparison only includes attributes and relations that cannot be changed at the instance. The comparison result is determined and displayed only for these attributes and relations.

Overview

The following table is an overview of the objects, attributes and relations that are relevant for comparison.

Object/type	Relevant attributes	Relevant relations
Equipment module		• Assigned block (block instance or sub-CFC) • Assigned name • Assigned comment • Assigned block icon
Configuring a sequential control	Change time stamp (of the assigned SFC type)	Assigned block (block instance or SFC type)
Control module		• Assigned block (block instance or sub-CFC) • Assigned name • Assigned comment • Assigned function identifier • Assigned block icon
Control module assignment (link object in the equipment module)		Control module assignment (not relevant for basic requirements)
Control variable	• Negation (only with interconnected inputs) • Name • Variable type (parameter/signal) • Data type • Comment	• Interconnection • Assigned I/O (block variable)
Message	• Name • Message identifier	Assigned message
Block instance	• Name • S7_mes • S7_read_back • Block type (function/function block)	Block type (instance of function/function block)

Object/type	Relevant attributes	Relevant relations
Block variable / block I/O	• Inverted (only with interconnected inputs) • Name • Comment • I/O type • Data type • Value (see the notes below). • S7_edit • S7_visible • S7_mes • S7_archive • S7_enum • S7_string0 • S7_string1 • S7_shortcut • S7_unit	Interconnection source
Sub-CFC	• Name • Version • Author	
CFC interface	-	-
CFC interface parameters	• Name • Comment • I/O type • Data type • Value (see the notes below). • S7_edit • S7_visible • S7_mes • S7_archive • S7_enum • S7_string0 • S7_string1 • S7_shortcut • S7_unit	
Block message	• Message identifier • Message type	
Sub-message	Message identifier (sub-message-number)	

Special notes

"Value" attribute

The "Value" attribute is ignored for the following objects under certain conditions:

- Block variables
- CFC interface parameters

The attribute is ignored if one of the following conditions exist:

- When there is an assignment to a control variable of the equipment module.
- When the "S7_m_c" attribute is set.

Under these conditions, the "Value" attribute is typically adapted to the instance.

Interconnections at the block instance

Interconnections at the block instance are also ignored if the "S7_contact" attribute is set at the block type. These interconnections are automatically made to the processing blocks, e.g., motor block, when the control module is assigned to the equipment module.

The "Value" attribute at this point is always ignored as well, because the parameters that are explicitly or implicitly (in the SFC type instance) defined as block contact are interconnected by definition.

See also

Comparing and synchronizing the type and instances of an equipment module (Page 286)

11.2.4.8 Attributes of the objects of an equipment module

Introduction

In the left section of the technological editor, the objects in a control module or equipment module are represented hierarchically in a tree structure, for example a "Parameter" or "Message".

The right section lists the attributes for the object that is currently selected in the tree.

Overview

The tables below show the attributes for the following objects of an equipment module:

1. Equipment module
2. Configuring a sequential control
3. Control module
4. Control module assignment
5. Control module (basic requirement)
6. Parameters

7. Signal
8. Message

Note regarding the "Data exchange" column in all subsequent tables

This column indicates whether an attribute is included in the data exchange with COMOS Integrated Engineering or Advanced ES.

1. Attributes of the "Equipment module" object

Attribute	Description of attribute value	Data exchange
Assigned chart	Cannot be edited	No
Name	Can be edited Not forwarded. Gets default value "CFC(x)".	Yes
Comment	Can be edited Not forwarded.	Yes
Author	Can be edited Not forwarded to associated objects.	Yes
Version	Can be edited	Yes
Sampling time	Can be edited All blocks of the assigned chart are installed in the OB with the most suitable sampling time. The next higher value will be entered in case of an invalid entry. The runtime editor is adapted accordingly.	Yes
Support type instance behavior	Can be edited The default setting for this attribute is TRUE for every instance of a technological type. This means that the instance is linked with its type and it is included in the type-instance synchronization. You can exclude instances selectively from the type-instance synchronization by deactivating this attribute. Deactivating this attribute for an instance will not eliminate its internal references and relation to the type. This allows you to edit the instance regardless of its type. You can include instances again in the type-instance synchronization by activating this attribute. Activating this attribute reestablishes the link between instance and its type thus including the instance in type-instance synchronization. Remark: • Deactivate this attribute only in exceptional cases, for example, testing, commissioning, selective, or step-by-step update of instances. • Keeping this attribute deactivated for a longer period might result in inconsistency of the instance with its type. Therefore, check the differences before synchronizing plant types.	Yes

Note

"Command" and "Status" attributes

No "Command" and "Status" objects can be defined in an equipment module. They can be defined either specifically for a control module type or basic requirement, or as "global" objects. You can find the description of the attributes of the "Command" and "Status" objects in the section "Attributes and attribute values of a control module (Page 214)".

2. Attributes of the "Sequential control" object

Attribute	Description of attribute value	Data exchange
Name	Can be edited Not forwarded. Gets the default value "Equipment phase".	Yes
Comment	Can be edited Not forwarded.	Yes
Author	Can be edited Not forwarded to associated objects.	Yes
Assigned block	Cannot be edited	No

3. Attributes of the "Control module" object

Attribute	Description of attribute value	Data exchange
Assigned block	Cannot be edited	No
Name	Can be edited Forwarded to associated objects. Gets the name of the assigned chart as default value. Can also be assigned to the CFC or a block.	Yes
Comment	Can be edited Forwarded to associated objects.	No
Operating icon	Can be edited	Yes
Optional	Can be edited Occurs only at the type. The attribute is not available for lower-level control modules, that is, for control modules that are directly below a type of a control module/ equipment module.	Yes
Author	Can be edited Not forwarded to associated objects.	No
Function identifier	Can be edited Forwarded to associated objects. No use for sub-control modules.	No

Attribute	Description of attribute value	Data exchange
Function	Can be edited Used to identify a control module as a function.	No
Function name	Displays the name of the function. • Cannot be edited at the type of the control module. When the "Function" attribute is enabled, the value of the "Name" attribute is applied automatically. • Can be edited at the instance.	No
Basic requirement	Can be edited Occurs only at the type. A function cannot be basic requirement.	Yes
Control module type	Cannot be edited Not forwarded.	No

There is no "Sampling time" attribute for control modules in equipment modules, because the control module is assigned in the runtime group of the equipment module.

4. Attributes of the "Control module assignment" object

Attribute	Description of attribute value	Data exchange
Assigned control module	Can be edited Not forwarded to associated objects.	Yes
Role	Can be edited Gets the default value "Role".	Yes
Comment	Can be edited Not forwarded.	Yes

5. Attributes of the "Control module (basic requirement)" object

Attribute	Description of attribute value	Data exchange
Assigned block	Cannot be edited	No
Name	Can be edited Not forwarded. Gets default value "Control module(x)".	Yes
Comment	Can be edited Not forwarded.	Yes
Operating icon	Can be edited	Yes
Optional	Can be edited Not forwarded to associated objects.	Yes
Author	Can be edited Not forwarded to associated objects.	Yes
Function identifier	Can be edited Forwarded to associated objects	Yes

Attribute	Description of attribute value	Data exchange
Function	Cannot be edited Used to identify a control module as a function.	No
Function name	Cannot be edited Displays the name of the function.	No
Basic requirement	Can be edited	Yes
Control module type	Cannot be edited Not forwarded.	No
Technological block	Can be edited Only displayed if there is no reference to a control module type.	No

6. Attributes of the "Parameter" object

Attribute	Description of attribute value	Data exchange
Assigned I/O	Cannot be edited	No
Interconnection to	Can be edited, Shows the interconnection.	No
Reference CM parameter	Can be edited Absolute or relative addressing possible.	Yes
Reference block variable	Can be edited Absolute or relative addressing possible.	Yes
Reference global variable	Can be edited Absolute or relative addressing possible.	Yes
Name	Can be edited Not forwarded Gets the default value "Parameter".	Yes
Comment	Can be edited Not forwarded.	Yes
Signal	Can be edited Right-click the text field to open the symbol table of the CFC using the menu command "New Signal Interconnection". The value is forwarded to the "Interconnection" property of the assigned I/O.	No
Value	Can be edited Forwarded to the "Value" property of the assigned I/O.	Yes
Negation	Select the check box to negate the control variable. The box can only be selected if there is an interconnection.	No
Text 0	Can be edited Forwarded to the "Text 0" property of the assigned I/O. This option is only visible when you are editing an I/O of the data type "BOOL" whose system attribute is set to `S7_string_0`.	Yes
Text 1	Can be edited Forwarded to the "Text 1" property of the assigned I/O. This option is only visible when you are editing an I/O of the data type "BOOL" whose system attribute is set to `S7_string_1`.	Yes

Attribute	Description of attribute value	Data exchange
Enumeration	Selection via drop-down menu Forwarded to the "Enumeration" property of the assigned I/O. This option is visible only when the I/O has the system attribute `S7_enum` assigned to it.	Yes
Unit	Selection via drop-down menu In addition to the text of the unit, a unique identification number according to the standard "Profile for Process Control Devices" is displayed in square brackets. Forwarded to the "Unit" property of the assigned I/O. The `s7_unit` attribute must be configured for this.	Yes
Operation identifier	Can be edited Forwarded to the "Identifier" property of the assigned I/O. The `s7_shortcut` attribute must be configured for this.	Yes
IO type	Selection via drop-down menu Must match IO type of the assigned I/O.	No
Data type	Selection via drop-down menu Must match data type of the assigned I/O.	Yes
Tag type	"Parameter" (selection via drop-down menu). Forwarded to the assigned I/O.	Yes
Control module type	Cannot be edited Not forwarded.	No

7. Attributes of the "Signal" object

Attribute	Description of attribute value	Data exchange
Assigned I/O	Cannot be edited	No
Interconnection to	Not relevant here.	No
Reference CM parameter	Not relevant here.	No
Reference block variable	Not relevant here.	No
Reference global variable	Not relevant here.	No
Name	Can be edited Not forwarded. Gets the default value "Signal".	Yes
Comment	Can be edited Not forwarded.	Yes
Signal	Can be edited Right-click the text field to open the symbol table of the CFC using the menu command "New Signal Interconnection". The value is forwarded to the "Interconnection" property of the assigned I/O.	No
Value	Not relevant here.	No

Attribute	Description of attribute value	Data exchange
Low scale value	Can be edited (at the type) Only relevant for analog signals (input and output signals) of "REAL" data type. With binary signals the input fields are locked.	Yes
High scale value	Can be edited (at the type) Only relevant for analog signals (input and output signals) of "REAL" data type. With binary signals the input fields are locked.	Yes
Negation	Select the check box to negate the control variable. The box can only be selected if there is an interconnection.	No
Enumeration	Not relevant here.	No
Unit	Not relevant here.	No
Operation identifier	Not relevant here.	No
IO type	Selection via drop-down menu Must match IO type of the assigned I/O.	No
Data type	Selection via drop-down menu Must match data type of the assigned I/O.	Yes
Tag type	"Signal" (selection via drop-down menu). Forwarded to the assigned I/O.	Yes
Control module type	Cannot be edited Not forwarded.	No

8. Attributes of the "Message" object

Attribute	Description of attribute value	Data exchange
Assigned message	Cannot be edited	No
Name	Can be edited Not forwarded to associated objects. Gets the default value "Message".	Yes
Message class	Cannot be edited This value is inherited from the "Message class" property of the assigned single message.	No
Priority	Can be edited Only numbers possible. The value "0" is entered in the case of an invalid entry. Forwarded to the "Priority" property of the assigned single message.	Yes
Message identifier	Must be specified manually. If a valid message identifier (e.g. SIG1) is entered, the message class, event and origin attributes are automatically applied as long as the assigned block has a message response.	No
Event	Cannot be edited Inherited from the "Event" property of the assigned single message.	No
Information	Forwarded to the "Infotext" property of the assigned single message.	No
Origin	Forwarded to the "Origin" property of the assigned single message.	No

"Assignment" column

Depending on the selected object in the left section of the technological editor, you can assign chart, blocks, block I/Os (parameter/signal), messages, control module, equipment module, or SFC type to the attributes of the selected object in the "Assignment" column.

You can assign blocks and parameters by dragging and dropping from the CFC editor. You can assign chart by selecting "Add Chart EM" menu command in the context menu of the selected field.

11.3 Configuring and using equipment phases

11.3.1 Basics of the equipment phase

Concept

As of PCS 7 V9.0, the "Equipment Phase" (EPH) is provided to enable control of multiple lower-level equipment modules.

- The equipment phase of the assignment partner is at the process control end when interfacing to SIMATIC BATCH.
- Equipment phases of various equipment modules and control modules can be stored for the process control.

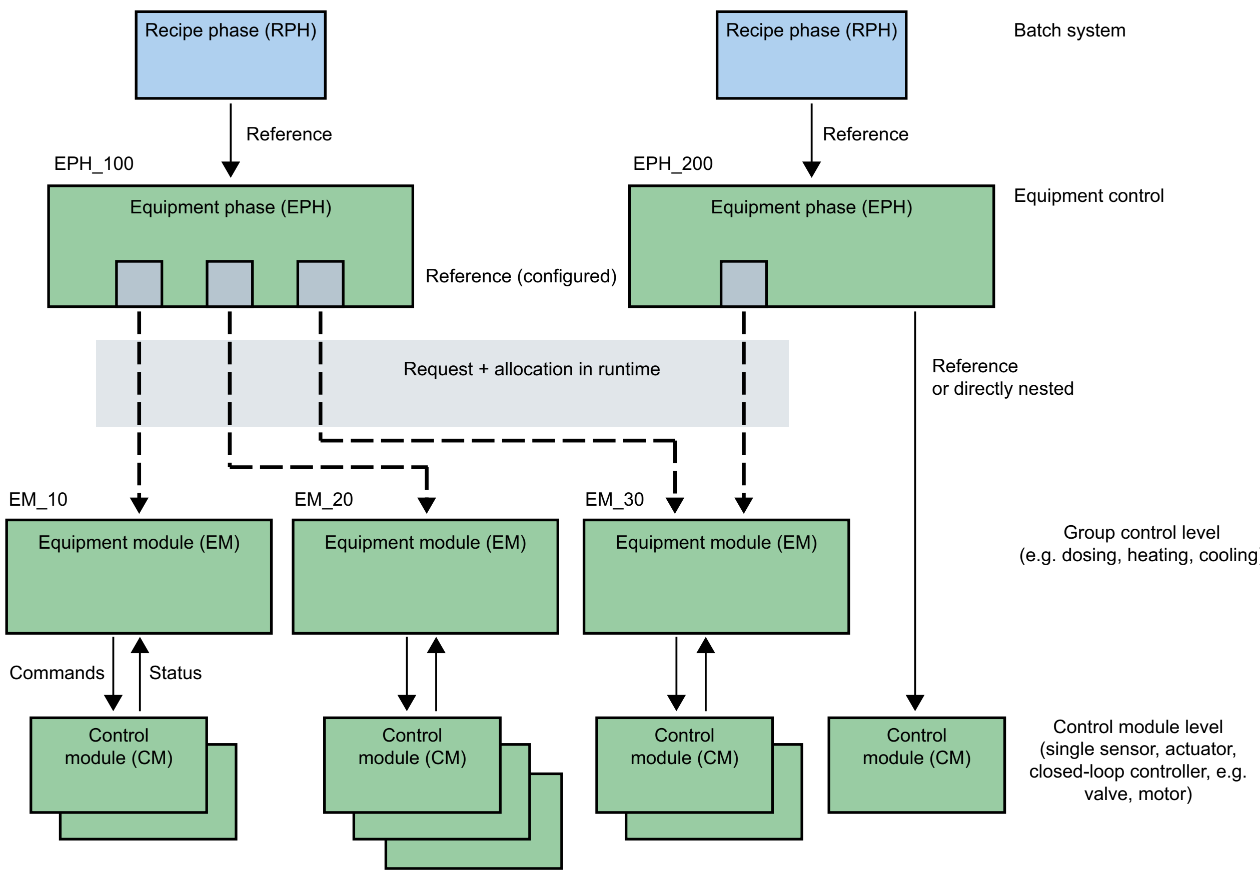
The advantages of the equipment phase are:

- The equipment phase is resource-neutral for SIMATIC BATCH. Therefore, the recipes in SIMATIC BATCH can be created independent of resources.
  Commands, statuses, control strategies and setpoints are sent, but no resource-specific data.
- For process control, the lower-level equipment modules (EM) are no longer permanently assigned, but instead requested at runtime, allocated and then released again.
  Because there is no more rigid assignment, lower-level equipment modules can be used by various equipment phases in succession. The strategy for the request, allocation and release of the lower-level equipment modules is configured in the equipment phase.

An equipment phase is represented by a faceplate in the process pictures of the PCS 7 OS and can also be operated through it in manual mode.

Overview

The figure below shows an example of the use of the technological objects, equipment phase, equipment module and control module.



Key:

- The equipment phase "EPH_100" uses the lower-level equipment modules "EM_10", "EM_20" and "EM_30".
- The equipment phase "EPH_200" uses the lower-level equipment module "EM_30".
  In addition, a control module (CM) is nested directly below it.
- In the "EPH_100" and "EPH_200" objects, the "References" are configured on the respective lower-level equipment modules. These "references" are symbolically represented in the picture and consist of the following configurations:
  – Objects for each lower-level equipment module:
    - "Equipment module (basic requirement)", which describe the minimum technological requirements for the equipment module.
    - "Equipment module assignment" (role), for an instance that is subsequently assigned to an equipment module.
  – Allocation strategy and, if needed, query of the allocation status (configuration in the sequential control of the equipment phase)
- The "Request + allocated at runtime" object shows that the equipment modules are requested in runtime, allocated and then released again.
  Because the "EM_30" object is required by both higher-level equipment phases "EPH_100" and "EPH_200", the query of the allocation status is useful here along with the configuration of the allocation strategy.

Basics

Elements of the equipment phase

The following description gives a brief overview of the data objects for an equipment phase.

You can find more detailed information on this in the section "Overview of the data objects of the equipment phase (Page 304)".

- The "Sequential control" with its sequencers and parameter descriptions always has to be available in the equipment phase, but only once.

The following objects can be optionally contained:

- Control module as an integral part of the type, several are possible
- Control module assignment; several are possible
- Control module requirements as basic requirements in which the minimum requirements for the control module are defined.
  Cannot be marked as optional; several are possible
- "Equipment module assignment"; several are possible
- "Equipment module (basic requirement)"; as basic requirement in which the minimum requirements for an equipment module are defined; several are possible
- The following objects are possible as control variables, several are possible:
  - Parameters
  - Signals
- Messages; several are possible

Elements of the control module

The control module can optionally contain the following objects:

- Sub control modules; several are possible
- Commands; several are possible
- Statuses; several are possible
- The following objects are possible as control variables, several are possible:
  - Parameters
  - Signals
- Messages; several are possible

Commands and status

The following applies to the equipment phase for "commands" and "status":

- Only global commands and status can be used for the equipment phase. These are created with the SIMATIC Manager in the master data library.
  These global commands and states are generally based on equipment phases and equipment modules and can be used in all equipment phases of a plant.
- Object-specific commands and status cannot be used in an equipment phase.
  This is only possible at the type of a control module (not at an instance) or at a "Control module (basic requirement)".

You can find additional information about this in the section "Overview of the data objects of global "Command" and "Status" (Page 352)".

The control module in the equipment phase

A control module can be assigned to an equipment phase.

The following configuration options are available:

- The control module to be assigned is integrated as a component of the equipment phase type and is therefore available as a lower-level object.
  For this, the control module is configured as an object directly in the equipment phase. A "Control module assignment" can also be configured in which the "Role" is defined.
- The control module is not a component of the equipment phase type.
  The "Control module (basic requirement)" and "Control module assignment" objects are then configured for this control module.

You can find additional information on this in the section "Configuration options of the control module in the equipment module / equipment phase (Page 256)".

Type and instance of an equipment phase

The type/instance concept is used in an equipment phase. The advantage of the type/instance concept is the ability to make changes to the type at a central location and then transfer the changes by synchronizing the instances.

- Synchronization is available when there are differences between the type and instance. Instance-specific extensions are managed as such and are not lost when the type and instance are synchronized. Changes can therefore be bumplessly loaded in the automation system.
- The type and instance of equipment phases are included in the data exchange with COMOS Integrated Engineering.

Creating a type and instance:

- A type of an equipment module can only be created in the master data library of the PCS 7 project using the following options:
  - In the "Plant view" in the SIMATIC Manager
  - By creating it in COMOS Integrated Engineering and subsequently importing it via the automation interface (AI)

  In the master data library, the type of equipment phase is configured for use in the PCS 7 project.
  The type can be copied and deleted in the master data library using the SIMATIC Manager.
- The instances are the repeated uses of a type in the plant description of the plant hierarchy. Instances of a type are created simply by copying the type from the master data library into the project under a node in the plant hierarchy.

Naming of type and instance

- The name of the type is unique within the master data library, just like the equipment module types and control module types.
- The types of equipment modules share a namespace in the master data library with the following objects:
  - Types of equipment modules
  - Control module types
  - Global commands and status

  Therefore, an equipment phase type cannot have the same name as a control module type.
- Name of an equipment module type:
  - The maximum permitted length of the name is 22 characters. It is recommended to use only 18 characters to provide some reserve for automatic renaming, for example, by the automatic addition of the character string "(1)" to avoid name collisions during copying.
  - The instances of the equipment phase type are hierarchically addressed through the plant hierarchy. Higher-level units are therefore unique.

Note

If the name already exists during copying:

If you copy a chart and paste it, the chart name suffixed with "(x)" (where x is 1, 2, 3, 4, and so on) is assigned as the name of the copied chart.

For example:

If there exists a chart named "ChartName" and if you create a copy of this chart, the copied chart is named as "ChartName(1)" because the name "ChartName" already exists. Further, if you paste it again (create another copy), the next copied chart is named as "ChartName(2)". But if you create a copy from the "ChartName(2)" chart then the copied chart will be named as "ChartName(2)(1)".

Note

If the maximum length of the name is exceeded during copying:

If you copy a chart which is named with 22 characters (maximum length) and paste it, the name of the copied chart is shortened from the end of the string until "~(x)" can be appended (where x is 1, 2, 3, 4, and so on).

For example:

If there exists a chart named "ChartNameMotorControlA" (22 characters) and if you create a copy of this chart then the copied chart is named as "ChartNameMotorCont~(1)".

By the general rules of naming in the case where the name already exists, the new chart name should have been "ChartNameMotorControlA(1)" (25 characters) but this exceeds the maximum permitted length of 22 characters. Therefore, the name is shortened to 22 characters with the methodology explained above.

Comparison and synchronization of type and instance

Synchronization of the type and the instances may be required when changes are made to the type or its associated instances.

During synchronization, all instances of a type in the project are compared to the corresponding type in the master data library. This shows the differences between the types of equipment phases and their instances and between the associated objects, such as CFCs (program level).

You can find additional information on this in the section "Comparing and synchronizing the type and instances of an equipment phase (Page 329)".

Data exchange with COMOS

The import functions of the automation interface (AI) can be used to transfer the types of equipment phases and equipment modules (basic functions) defined on the engineering level with COMOS Integrated Engineering to the PCS 7 process control system. The associated sequencers, transition conditions and actions are imported along with the types.

This enables the data generated on the engineering level in COMOS in the form of equipment phases and equipment modules to be imported into PCS 7 and thereby entered in the automation layer. There, the data appear as sequencers (SFC) and automation programs (CFC).

You can find additional information on this in the section "Overview of data exchange with COMOS (Page 332)".

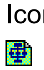
See also

Configuration methods for assigning a recipe phase (RPH) to the engineering system (Page 345)

11.3.2 Overview of the data objects of the equipment phase

Introduction

The following tables describe:

1. Data objects of the "Equipment phase"
   Icon in the technological editor:
   
2. Data objects of the "Sequential control"
   Icon in the technological editor:
   

Data objects of the equipment phase

An equipment phase (EPH) consists of the following elements, of which there can be one or several.

Note

No "Command" and "Status" objects can be defined in an equipment phase. They can be defined either specifically for a control module type or basic requirement, or as "global" objects. You can find more information on this in the tables below.

Object/icon	Description
Configuring a sequential control	The sequential control defines the sequencers and their parameter descriptions (e.g. setpoints, control strategies, parameters, messages). The sequential control may only be in the equipment phase once. Each equipment phase must be assigned one and only one sequential control. The elements are described in the table below, "Data objects of the sequential control".
Control module	A control module can be configured as component of an equipment module type. Several control modules can be present. Optional control modules: • Control modules can only be defined as "optional" at the type in the master data library. • Control modules can only be defined as "optional" on the first sub-level under an equipment module type. • The nesting depth for control modules under the equipment module is limited to a maximum of 3 levels. This means that an equipment module can only have a maximum of two sub-levels of sub-control modules, for example, the levels "Equipment module > Control modules > Sub-control modules". **Note:** A control module that is a component of the equipment module type can be defined as a "basic requirement". The "Basic requirement" attribute is set at the control strategy for this. The control module is then no longer a component of the equipment module type and the blocks of the control module are deleted from the model.

Object/icon	Description
Control module assignment 	If a control module is to be controlled in the sequencers, the equipment module must be assigned to the control module. The assignment configures the following: 1. The "role". The so-called "role" of the control module is set in the equipment module type, but no specific control module. The "role" indicates how the assigned control module is used in the context of this type, e.g. that an assigned "valve" control module has the "role" of an "outlet". The "role" is used to address the target control module in the actions and transition conditions of the sequencers. 2. The link to a control module or basic requirement. – When the control module is integrated in the type of the equipment module, this control module is assigned. – If the control module is not integrated in the type, a corresponding basic requirement is assigned. The specific requirements for the control module to be assigned are specified themselves under "Control module (basic requirement)" described below. The assignment can be provided with a comment. There can be more than one assignment. If a "Control Module Assignment" is configured, you must also create a corresponding basic requirement or a control module that is integrated in the type.
Equipment module assignment 	The assignment configures the following: 1. The "role". The type of the equipment module defines the "role" of an equipment module, but no specific equipment module. The "role" indicates how the assigned equipment module is used in the context of this type, for example, that an assigned "valve" equipment module has the "role" of an "outlet". The "role" is used to address the target equipment module in the actions and transition conditions of the sequencers. 2. The link to an equipment module or basic requirement. – If the linked equipment module is configured in the type of equipment module, this equipment module is assigned. – If the linked equipment module is not configured in the type, a corresponding basic requirement is assigned. The specific requirements for the equipment module to be assigned are specified themselves under "Equipment module (basic requirement)" described below. The assignment can be provided with a comment. There can be more than one assignment. If an "Equipment module assignment" is configured, you must also create a corresponding basic requirement or a lower-level equipment module that is integrated in the type. Remark: If the interconnections between EM and EPH instances are not updated after executing "Update block types" on SFC types of EM and EPH, you must reassign the EM role in the EPH instance.

Object/icon	Description
Control module (basic requirement)	The minimum requirements for the control module to be assigned are abstractly described in the "Control module (basic requirement)". The requirements include, in particular, the necessary commands, but also feedback about the status. • There can be more than one basic requirement. • The basic requirements are needed at the equipment module type to enable the use of commands and states, for example, in the procedural logic. • The "Command" and "Status" objects can be defined at the basic requirement. You can find additional information on this in the section "Special considerations for the implementation of specific commands/statuses (Page 213)". • A basic requirement cannot be defined as optional.
Equipment module (basic requirement)	The minimum requirements for the equipment module to be assigned are abstractly described in the "Equipment module (basic requirement)". The requirements include, in particular, the necessary commands, but also feedback about the status. • There can be more than one basic requirement. • The basic requirements are needed at the equipment module type to enable the use of commands and states, for example, in the sequential logic. • The "Command" and "Status" objects can be defined at the basic requirement. You can find additional information on this in the section "Special considerations for the implementation of specific commands/statuses (Page 213)". • A basic requirement cannot be defined as optional.
Parameters	• There can be several parameters. • The "Parameter" object can be used: – As a sub-object of an equipment module or control module. – As a formal parameter of commands and statuses. In the "Sequential control", these formal parameters define placeholders for specific parameters or constants at the points of use in the actions and transitions of the sequencer. • The "Value" and "Signal" attributes and the "Interconnection" relation can only be entered as alternatives. • The "Signal" attribute is only allowed for input parameters and is disabled for output parameters. • The "Negation" attribute is only allowed for input parameters of the "BOOL" data type and is disabled for output parameters. For input parameters, the "Negation" attribute relates to the parameter interconnection or the signal.

Object/icon	Description
Signals	• There can be several signals. • The attributes of signals are represented in the same way as parameters in the technological editor. The input fields of the "Value", "Enumeration", "Operation identifier", "Unit" and "Interconnection" values or relations are disabled, however. • The "Negation" attribute is only allowed for input parameters of the "BOOL" data type and is disabled for output parameters. It applies to the "Signal" attribute for signals.
Messages	An equipment module can generate messages by default. • There can be several messages. • Messages can be configured directly under the equipment module. They then apply only to the block instance of the SFC type in the associated CFC. To assign the corresponding message, the message inputs of this block instance of the SFC type must be set as visible.

Data objects of the sequential control

The "Sequential control" consists of the following elements, of which there can be one or several.

Object/icon	Description
Parameters Used for the parameter description	• The parameter descriptions of the "Sequential control" correspond to the "characteristics", which are set at the SFC type. • In the context of integration on the engineering level (COMOS Integrated Engineering), there is a limitation to the parameters and attributes required for the process engineering in the data exchange models. • Parameters can be used as sub-objects of a "Sequence control" to define as individual variables the "Sequence control" interface externally and internally to the sequencers. You can find more information on this at the end of this table.
• Control strategies	Various process engineering procedures can be defined using control strategies. The control of sequencers is basically a function of the control strategies. Control strategies are important for higher-level recipe control level (batch systems). Attributes: • Name • Number • Comment • Assigned setpoint The assignment of a setpoint to a control strategy is performed in the COMOS data exchange model using a relation (assigned control strategy).

Object/icon	Description
• Setpoints	Setpoints can be specified in process engineering by manual operation or by a higher control level (batch system). Setpoints are assigned to individual control strategies. The actual (process) value is always offered as a control variable in addition to the setpoint. Attributes: • Name • Data type • Comment • Low limit • High limit • Unit High, low limit and unit (of measure) are typically determined or adapted at the instance of the equipment module.
• Process values / actual values	The integration of process values in the equipment module is used to control sequencers, e.g., using an actual value for step-enabling the sequencers. Attributes: • Name • Data type • Comment • Unit The unit is typically set at the instance of the equipment module.
• Control values	Control values are required to connect external control modules, which are not directly accessible through the control module contacts. Attributes: • Name • Data type • Comment • Unit The unit is typically set at the instance of the equipment module.
• Parameters	A parameter is used to influence the behavior of a "Sequential control" in an instance, for example, with options. These parameters can also be used to set limits for specific instances. Attributes: • Name • Data type • Comment • Unit
• Bit memory	Bit memory is needed to temporarily store values. It is only used locally in the sequential control system. Attributes: • Name • Data type • Comment

Object/icon	Description
• Timers	Timers are used, for example, for time monitoring or calculation of run time. Attributes: • Name • Comment
• Note texts	Note texts are used, for example, for the user interface. Attributes: • Name • Text
• Position texts	Position texts are used to display the current procedural status on the operator station. Attributes: • Name • Text
Sequencer	The process engineering task – the "sequential control" – is defined in a sequential logic. This is described using sequencers. The behavior must be defined for each control strategy for each state of the "Sequential control". The sequencers have exactly one start step and one end step. Sequencers can include the following elements, which are familiar from the SFC type: • Steps • Transitions • Alternative branches • Simultaneous branches • Loops • Jumps Both alternative and simultaneous branches are symmetrically merged again. The actions that are defined for a step on the process engineering level in COMOS are described with commands in the form of a list. The commands used for this are provided centrally on the control module, for example, motor, valve. Commands can be defined as well at the equipment module. The transition conditions are set on the process engineering level in COMOS as expressions with a Boolean result using the status and logical standard functions. For this, the status (or states) provided by the control modules, for example, motor, valve, can be used as well as the status defined at the equipment module.

Parameters as individual variables for step and transition configuration

In a sequencer, various individual variables, e.g. actual values, limit values may be required for step and transition configuration.

To create these individual variables, the required block I/O of the block instance of the corresponding SFC type is selected and dragged to the "Sequential control" in the technological editor. A "Parameter" object is automatically created. The names of these individual variables are pre-defined and cannot be changed.

These individual variables are interconnected and assigned parameters at the instance of the equipment module.

Data objects of the control module

One or more control modules can be assigned to an equipment phase.

In connection with the equipment phase, it is particularly important that control modules be accessible via commands and states of the equipment phase sequencers.

You can find a description of the data elements of the control module in the section "Overview of data objects of the equipment module and control module (Page 269)".

See also

Basics of the equipment phase (Page 299)

Configuring a global command or status (Page 348)

11.3.3 Configuring and managing equipment phases

11.3.3.1 Overview for configuring and managing the equipment phase

Overview

- Configuring
  A type of an equipment phase can only be created in the master data library of the PCS 7 project using the following options:
  – In the "Plant view" in the SIMATIC Manager
  – By creating it in COMOS Integrated Engineering and subsequently importing it via the automation interface (AI)

  In the master data library, the type of equipment phase is configured for use in the PCS 7 project.
  You can find additional information on this in the section "Configuring an equipment phase (type) (Page 312)".
- Management
  An equipment phase type can be copied and deleted in the SIMATIC Manager.
- Create an equipment phase with "derived interface"
  You can find additional information on this in the section "Create an equipment phase with "derived interface" (Page 314)".
- Configuring the sequential control of the equipment phase
  You can find additional information on this in the section "Configuring the sequential control of an equipment phase (Page 318)".
- Configuring lower-level equipment modules
  Equipment modules can be assigned as lower-level objects in the equipment phase.
  You can find additional information on this in the section "Configuring a lower-level equipment module (Page 320)".

- Configuring the allocation of a lower-level equipment module
With equipment phases, the lower-level equipment modules are not permanently assigned, but instead requested at runtime, allocated and then released again.
You can find additional information on this in the section "Configuring the allocation of an equipment module (Page 322)".
- Control modules on the type of equipment phase
Control modules can be assigned as lower-level objects in the equipment phase or integrated into the equipment phase.
You can find additional information on this in the section "Configuration options of the control module in the equipment module / equipment phase (Page 256)".
- Creating an instance
An instance can be created by copying the equipment phase type from the master data library into a node of the plant hierarchy.
You can find additional information on this in the section "Creating an instance of the equipment phase type (Page 327)".
- Type/instance synchronization
Synchronization of the type and the instances may be required when changes are made to the type or its associated instances.
You can find additional information on this in the section "Comparing and synchronizing the type and instances of an equipment phase (Page 329)".
- Control modules at the instance of the equipment phase
If the control module is configured at the type as a "Control module (basic requirement)", a specific control module from the project must be assigned at the instance.
You can find additional information about this in the section "Assigning the control module instance to an equipment module/phase (Page 262)".

11.3.3.2 Configuring an equipment phase (type)

Introduction

An equipment phase type is only created in the master data library of the PCS 7 project using the following options:

- In the "Plant view" in the SIMATIC Manager
- By creating it in COMOS Integrated Engineering and subsequently importing it via the automation interface (AI)
You can find additional information about this in the section "Overview of data exchange with COMOS (Page 332)".

The type can be copied and deleted in the master data library using the SIMATIC Manager.

In the master data library, the type of equipment module is configured for use in the PCS 7 project.

This procedure is described below.

Requirements

- A multiproject containing a master data library is open in the SIMATIC Manager.

Procedure

1. Use the menu command "View > Plant View" to open the plant view.
2. Once the equipment phase is imported, proceed to step 4.
3. If you wish to recreate an equipment phase type, in the plant view of the master data library navigate to the directory in which the object is to be created, for example, "Process tag types".
   Select the "Insert New Object > Equipment Phase (Type)" command from the shortcut menu.
   Proceed to step 5.
4. In the plant view of the master data library, navigate to the directory where the equipment phase was created or stored after import, for example, "Process tag types".
   You can move or copy the type into another directory you have created in the master data library and configure the properties there.
5. The equipment phase type appears in the right window with the corresponding icon.
   
6. Double-click on the icon. The CFC editor opens; this is where the properties of the equipment phase type are configured.
   Open the technological editor. The type has different attributes, such as "Name", "Author", "Comment".
   Enter the desired data for the attributes.
7. If you want to configure lower-level elements under this type, select the icon for the type of the equipment phase in the technological editor.
   

8. Select the "Insert New Object" command from the shortcut menu.
   All available elements are shown in the shortcut menu.
   - "Control module"; for a sub-control module that is a component of the equipment module
   - "Control module (basic requirement)"; as basic requirements in which the minimum requirements for the control module are defined
   - "Control module assignment"
   - "Equipment module assignment"
   - "Equipment module (basic requirement)"; as basic requirement in which the minimum requirements for an equipment module are defined
     You can find information about configuration of the "Equipment module assignment" and "Equipment module (basic requirement)" objects in the section "Configuring a lower-level equipment module (Page 320)".
   - "Parameter"
   - "Signal"
   - "Message"
   - "Sequential control"
     The sequential control may only be in the equipment phase once. Therefore, the menu command is enabled only when no "Sequential control" object is currently present in this equipment phase. You can find information about configuration in the section "Configuring the sequential control of an equipment phase (Page 318)".

   Select the appropriate menu command to insert the desired object, for example, "Parameter".
9. Repeat this step as often as necessary to add more lower-level elements.

Result

The equipment phase type is configured.

Lower-level elements such as parameters and control modules are inserted.

Note

The objects, for example, the values of attributes or assignments still have to be configured in the technological editor.

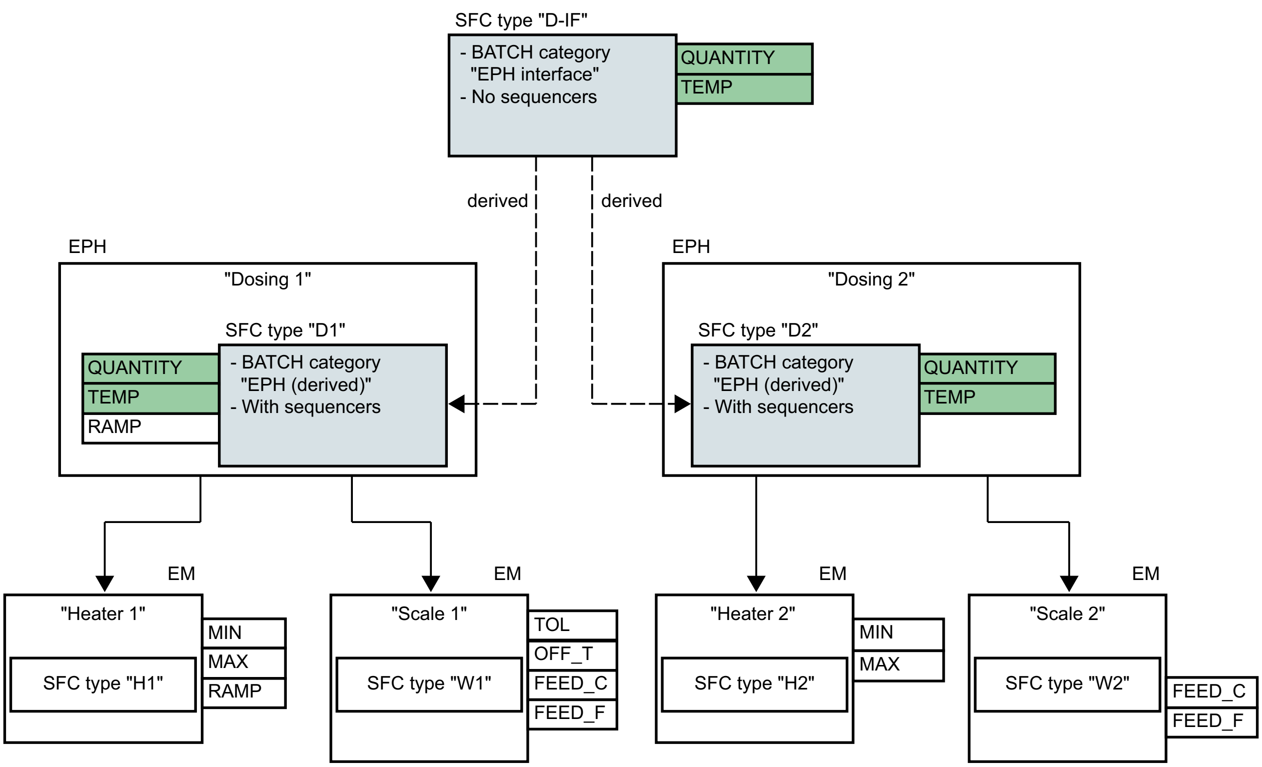
11.3.3.3 Create an equipment phase with "derived interface"

Introduction

Principle

The figure below shows the principle for using SFC types with the BATCH categories "EPH interface" and "EPH (derived)". Equipment phase

With this method of configuration, the SFC type with "EPH (derived)" category serves as an assignment partner for the recipe phase (RPH) in SIMATIC BATCH, which is derived from an abstract interface in SFC type with the "EPH interface" category.



Key:

- The BATCH category "EPH interface" is configured in the SFC type "D-IF".
  This SFC type has no sequencers as an "EPH interface".
- The associated SFC types "D1" and "D2" are configured for the EPH "Dosing 1" and "Dosing 2".
  - The BATCH category "EPH (derived)" and the name of the "D-IF" interface from which these two SFC types are derived is configured in the SFC types "D1" and "D2".
  - The sequencers can be configured in these SFC types as SFC type with the "EPH (derived)" category.
  - As derivatives of "D-IF" they have at least the same parameters, control strategies, setpoints, for example, "QUANTITY", "TEMP", but may also contain additional and different parameters, for example, "RAMP".
- The EM "Heater 1 + 2" and "Scale 1 + 2" are lower-level EMs below the EPH "Dosing 1" and "Dosing 2" with separate SFC types.

"EPH interface"

- The BATCH category "EPH interface" is set in the properties of the SFC type of the "EPH interface".
- An SFC type with the BATCH category "EPH interface" serves as a basic class and thus as a generic interface for the integration of the recipe phase (RPH) of SIMATIC BATCH.
- The SFC type with the BATCH category "EPH interface" only contains an interface description with setpoints, control strategies, etc. but no implementation. The programming of sequencers with the steps and transitions is no longer possible in the SFC editor in this case.

"EPH (derived)"

- The BATCH category "EPH (derived)" is set and the name of the SFC type of the "EPH interface" is configured in the properties of the SFC type of "EPH (derived)".
- An SFC type with the BATCH category "EPH (derived)" contains the interface description with setpoints and control strategies as well as an implementation. The programming of the sequencers in the SFC editor is possible.
- Several "EPH (derived)" SFC types can be derived from the same "EPH interface" SFC type. The interfaces of these derived "EPH (derived)" SFC types need not be identical, but they must fulfill the requirement of the "EPH interface" SFC type.

Requirements

- A multiproject containing a master data library is open in the SIMATIC Manager.
- The master data library contains an equipment phase (EPH) whose interface is to be derived.

Procedure

1. Use the menu command "View > Component View" to change to the component view.
   Navigate through the master data library to the "Charts" directory.
2. Select the "Insert New Object > SFC Type (EPH)" command from the shortcut menu.
   The newly created SFC type appears in the "Charts" directory.
   If needed, change the name of the newly created SFC type, for example, to "D-IF".
3. Select the newly created SFC type and select the menu command "Object properties" in the shortcut menu.
   The "SFC Type Properties" dialog box opens.
   Switch to the "Options" tab.
4. In the "Category" selection box of the "SIMATIC BATCH" area, select the option "EPH interface".

5. Configure the desired setpoints, control strategies, etc. in the characteristics of the SFC type.
   Close the SFC editor.
   The SFC type with BATCH category "EPH interface" is displayed with the corresponding icon in the "Charts" directory.

6. Switch to the plant view and open the desired equipment phase whose interface is to be derived from the previously created SFC type, for example, "Dosing 1".
7. In the CFC of the equipment phase, mark the SFC block and select the command "Open SFC Type" in the shortcut menu.
   The associated SFC type is displayed in the SFC editor, for example, "D1".
8. Select the menu command "SFC > Properties".
   The "SFC Type Properties" dialog box opens.
   Switch to the "Options" tab.
9. In the "Category" selection box of the "SIMATIC BATCH" area, select the option "EPH (derived)".
10. Select the desired interface, for example, "D-IF", in the "Derived from interface:" field. All available SFC types with the BATCH category "EPH interface" are displayed for selection in the field.
    Click "OK".
    The SFC type with the BATCH category "EPH (derived)", for example, "D1", is opened in the SFC editor.
    The SFC type is displayed with the corresponding icon in the "Charts" directory.

11. Also open the SFC type with "EPH interface" category from which the interface is derived, for example, "D-IF".
    To do this, select the menu command "Open SFC type".
12. Open the editor window for the characteristics for both SFC types. Click on the "Characteristics" icon in the toolbar.
13. Check the characteristics in both SFC types.
    The characteristics of the SFC type with the "EPH interface" category should all be available in the SFC type with the category "EPH (derived)".

Result

An SFC type with the BATCH category "EPH interface" with the interface description with setpoints and control strategies is created and serves as a basic class.

An SFC type with the category "EPH (derived)" was created with derived interface from this SFC type with the "EPH interface" category.

This SFC type with the "EPH (derived)" category is part of a sequential control of an equipment phase.

Note

Only the configuration of the SFC types with the BATCH categories "EPH interface" and "EPH (derived)" is included in the above description.

However, additional steps are needed for complete configuration; for example, additional parameters and the sequencers in the SFC type with the BATCH category "EPH (derived)" must be implemented.

See also

Configuration methods for assigning a recipe phase (RPH) to the engineering system (Page 345)

11.3.3.4 Configuring the sequential control of an equipment phase

Introduction

The sequential control may only be in the equipment phase once.

- When a new equipment phase is created, a "Sequential control" object and an associated SFC type are created automatically. This assigned SFC type defines the sequencers and their parameter descriptions (for example, setpoints, control strategies, parameters, messages).
- If the "Sequential control" object is deleted in the equipment phase and added again, an SFC type must once again be assigned to the object.
- The SFC type that is assigned to a "Sequence control" object cannot be deleted in the "Charts" directory of the master data library as long as the associated "Sequence control" object exists in an equipment phase.

Requirements

- A multiproject containing a master data library is open in the SIMATIC Manager.
- An equipment phase type is available in the master data library and configured as described in the section "Configuring an equipment phase (type) (Page 312)".

Procedure

1. In the master data library, open the equipment phase type in the CFC editor.
2. Select the icon for the type of equipment phase in the technological editor.

   All known objects are displayed under this icon.
   In a newly created equipment phase, the "Sequential control" object and an associated SFC type are automatically available.
   If the "Sequential control" object already exists, check in the technological editor if there is an entry in the "Assigned" column of the "Assigned block" attribute. If there is an entry and an SFC type is therefore assigned, proceed to step 6.
3. If the "Sequential control" object is not available, select the "Insert New Object" command from the shortcut menu. All available objects are shown in the shortcut menu.
   Select the "Sequential control" menu command.
   The sequential control is inserted and displayed as an icon.

   Because each equipment phase can be assigned one and only one sequential control, the corresponding menu command is disabled in the shortcut menu of the equipment phase afterwards.
4. Select the icon for the "Sequential control" in the technological editor. The right window shows the associated attributes.
   Select the "Assigned block" attribute. When creating the sequential control manually, there is no entry in the "Assignment" column and thus no assigned SFC type.
5. To assign an SFC type, open the block catalog in the CFC editor.
   Drag the desired SFC type from the block catalog and drop it in the chart window of the CFC editor. The newly inserted SFC type is displayed in the CFC and referenced in the "Assigned block" attribute.
   If no SFC type appears in the block catalog, create the desired type in the "Charts" folder of the master data library in the component view.
6. If you want to configure lower-level elements for this sequential control, select the icon for the sequential control in the technological editor.
   Select the "Insert New Object" command from the shortcut menu. All available objects are shown in the shortcut menu. Only "Parameter" can be selected in the current version.

7. Select the "Parameters" menu command in the shortcut menu.
   The "Parameters of the equipment phase" dialog opens. All the parameters that can be used are shown in this dialog.
   Select the required parameter and click "Insert". The parameter is inserted and displayed as an icon. Configure the attributes of this parameter.
8. Parameters can also be used as individual variables for configuring steps and transitions, e.g., for actual value, limit values.
   If you would like to create a parameter as an individual variable, select the desired block I/O of the block instance of the corresponding SFC type and drag it to the sequential control in the technological editor.
   The "Parameter" object is created automatically. The names of these individual variables are pre-defined and cannot be changed. You can find additional information in the section "Overview of the data objects of the equipment phase (Page 304)" in the table of the data objects of a sequential control.

Result

The "Sequential control" object is configured at the type of an equipment phase and an SFC type is assigned.

Parameters are created as subobjects of this sequential control, if required.

Note

For the assigned SFC type, you still need to configure the contents in the SFC editor, for example, the sequencer and characteristics.

11.3.3.5 Configuring a lower-level equipment module

Introduction

The assignment of a lower-level equipment module to an equipment phase is configured in the technological editor in the equipment phase.

The following objects are configured to use the equipment module as a lower-level object of an equipment phase:

- An "Equipment module (basic requirement)" object, represented by the following icon:

  The minimum technological requirements for the equipment module to be assigned are described in the basic requirement.
- An "Equipment module assignment" object, represented by the following icon:

  The corresponding basic requirement is assigned in the "Equipment module assignment" and the "Role" is configured for the instance of an equipment module to be later assigned.

The following section describes this configuration.

Requirement

An equipment phase type is created in the master data library.

Procedure

1. Open the desired equipment phase type in the CFC editor.
2. Select the icon for the type of equipment phase in the technological editor.

   
3. Select the "Insert New Object > Equipment Module (Basic Requirement)" command from the shortcut menu.
   The "Equipment module (basic requirement)" is inserted and displayed as an icon.

   
4. Select the icon for the newly created "Equipment module (basic requirement)" in the technological editor.
   The associated attributes are displayed.
   Configure the attributes, for example, name, author.
5. Select the icon for the type of equipment phase in the technological editor.
6. Select "Insert New Object > Equipment Module Assignment" from the shortcut menu.
   The "Equipment module assignment" is inserted and displayed as an icon.

   
7. Select the icon for the newly created "Equipment module assignment" in the technological editor. The associated attributes such as "Interconnection to" and "Role" are displayed.
   Because this associated equipment module is not part of the type, a basic requirement is configured for the "Equipment module assignment" attribute.
   To configure the "Interconnection to" attribute, drag the basic requirement created above to the "Equipment module assignment" object in the technological editor.
   The reference you have just created to the basic requirement is displayed in the "Interconnection to" attribute.

8. Configure the "role" for the attribute with the same name.
   The "role" indicates how the associated basic requirement is to be used in the context of this type.
   Enter a comment, if needed.
9. If you want to configure lower-level elements under the "Equipment module (basic requirement)", for example, the "Status", select the icon for the "Equipment module (basic requirement)" in the technological editor.
   
10. Select the "Insert new object" command from the shortcut menu.
    All available elements are shown in the shortcut menu.
    – "Control module"
    – "Equipment module (basic requirement)"
    – "Parameters"
    – "Signal"
    – "Message"
    – "Control module assignment"
    – "Equipment module assignment"

    Select the appropriate menu command to insert the desired object, for example, "Parameter".
    Repeat this step as often as necessary to add more lower-level elements.
11. Configure the attributes of the lower-level elements.

Alternative procedure for steps 2 to 4

If the equipment module to be placed under the equipment phase already exists in the master data library, a corresponding basic assignment can be created using this equipment module.

- Select the icon of the equipment module in the template catalog of the CFC editor.
- Drag the icon and drop it on the "Equipment phase" object in the technological editor.
  The "Equipment module (basic requirement)" object is created automatically. The name is identical to the name of the equipment module used.

Result

The "Basic requirement" and "Assignment" of an equipment module are configured on the type of an equipment phase.

This equipment module is abstractly described in the base requirement.

11.3.3.6 Configuring the allocation of an equipment module

Introduction

With equipment phases, the lower-level equipment modules are not permanently assigned, but instead requested at runtime, allocated and then released again.

There is no more rigid assignment and lower-level equipment modules can be used by various equipment phases in succession.

The allocation characteristics are configured using an "allocation request" in the step.

The current status of the allocation at runtime is shown with the "Allocation status" and can be evaluated in a transition.

Allocation request and allocation status

- The allocation requirements are defined in the master data library under "Shared Declarations > Enumerations > ES_AcquireTypes".
  The following table describes the available allocation requests sorted according to their priorities.
  The lowest priority is at the top.

Allocation request (Type according to ES_AcquireTypes)	Priority	Network share	Description
No request (NoRequest)	0	-	Default value of the allocation variable Is also used to withdraw an allocation request.
Preliminary (wait) (PrelimWait)	1	Implicit	Similar to "Wait", only with implicit release in a subsequent, higher-level allocation request.
Preliminary (now) (PrelimNow)	2	Implicit	Allocation must be immediately successful or it fails. Implicit release for a subsequent, higher-level allocation requirement.
Wait (Wait)	3	Explicit	If no immediate allocation is possible, there is a wait for the release. If the passed wait time is zero, the wait may last indefinitely. This allocation request fails when a wait time greater than zero has been configured and this time has expired. Explicit release required.
Immediate (Now)	4	Explicit	Allocation must be immediately successful or it fails. Explicit release required.
Takeover (Takeover)	5	Implicit	With the same allocation identifier (for example, unit name), an immediate change of ownership takes place. In regard to errors and wait time; otherwise like the "Wait" type.
Force (safe) (ForceSafe)	6	Explicit	Once the equipment module (EM) to be allocated is at an end position (IDLE, ABORTED, etc. – ORing possible as well) that can be determined by parameter, it changes ownership.
Force (now) (ForceNow)	7	Explicit	Immediate, unconditional change of ownership by the requesting object.

- The allocation states are defined in the master data library under "Shared Declarations > Enumerations > ES_AcquireStates".
  The following table describes the available allocation states.

Allocation status (label according to ES_AcquireStates)	Comment
Idle	Free Equipment module (EM) is available.
Owner	Owner Allocation has succeeded.
Waiting	Waiting Request was accepted, but equipment module (EM) is not yet available.
NotAvailable	Not available Equipment module (EM) is not available at the time of the request. Can only occur with the allocation requests "Now (Now)" and "Preliminary (now) (PrelimNow)".
Deactivated	Deactivated The requested equipment module (EM) can generally not be allocated according to its own information.
Timeout	Maximum wait time has expired
Snatched	"Snatched" Equipment module (EM) has been forcibly taken over by another location.
ConfigError	Configuration error Only applies to the allocation type "Force (safe) (ForceSafe)". Occurs when the description masks for a safe state do not match, for example, bits should be included per OR logic operation and simultaneously be excluded per AND_NOT logic operation.

Requirement

Equipment phases and equipment modules are configured as types in the master data library.

Procedure

1. Open the required type of the equipment phase. Navigate to the "EPH" block in the CFC of the equipment phase and select the menu command **Open SFC Type** in the shortcut menu. The SFC type is opened in the SFC editor.
2. Double-click on the desired step or select the menu command **Edit > Object Properties....** for the highlighted step. The "Properties" dialog box for the step opens.
   Switch to the "Actions (technological)" tab.
3. Click on the "Action (technological)" button of the desired table row. The "Action (technological)" dialog box opens.
4. Select the top cell in the left column of the table and select the menu command **Insert Command** from the shortcut menu.

5. From the menu, select the desired equipment module for which the allocation requirement is to be configured.
   In the subsequent menu, select the menu command "EM_Allocate" and "EMAQAcquire" in the submenu.
   The action selected in the "Action (technological)" dialog box is inserted under the name "<Object name>.EMAQAcquire".
6. From the newly created action "<object name>.EMAQAcquire", select the subobject "Strategy".
7. Select the desired request behavior, for example, "PrelimWait" via a selection menu in the corresponding cell of the "Interconnection" column.
   You can find more information on the available options in the description above.
8. Click the "Apply" button to save the changes made to the instructions of the highlighted step.
9. Click "Close". The dialog box for editing the highlighted step closes. If you have made changes and not saved them, a prompt appears.

Result

The allocation request for a lower-level equipment module is configured in an equipment phase.

Status evaluation of the allocation request

To evaluate the success of the allocation request, a technological condition can be configured in a subsequent transition in the sequencer of the equipment phase.

A status is inserted in this condition and the desired equipment module and status are selected according to the table above.

11.3.3.7 The control module in the equipment phase

Overview

A control module can be assigned to an equipment phase.

The following configuration options are available:

- The control module to be assigned is integrated as a component of the equipment phase type and is therefore available as a lower-level object.
  For this, the control module is configured as an object directly in the equipment phase. A "Control module assignment" can also be configured in which the "Role" is defined.
- The control module is not a component of the equipment phase type.
  The "Control module (basic requirement)" and "Control module assignment" objects are then configured for this control module.

You can find additional information about this in the section "Configuration options of the control module in the equipment module / equipment phase (Page 256)".

11.3.3.8 Creating an instance of the equipment phase type

Overview

The equipment phase type is stored in the master data library.

The type can be copied within the master data library or copied to create an instance in the project.

Properties when copying an equipment phase type in the master data library

- When a type is copied within the master data library, a new type is created and all the components of the original type, including the basic requirements, are copied along with it.
- A "Control module assignment" is copied as well, regardless of whether the assigned control module therein is a component of the type or a basic requirement.
- A defined "Equipment module assignment" in the type is also copied.
  No equipment module can be directly integrated in the equipment phase; this means there is only a basic requirement for the assigned equipment module.

Properties when creating an instance (copying from the master data library into the project)

- An instance can be created by copying the equipment phase type from the master data library into a node of the plant hierarchy.
  This node can be classified as an equipment phase according to the ISA-88 standard.
  If SIMATIC BATCH is used, however, this node must be labeled as an equipment phase.

  Note

  Number of instances in a process cell node

  Any number of instances of an equipment phase type can be created under a process cell node in the plant hierarchy. If this node is classified as an equipment phase according to ISA-88, however, multiple instances do not make sense. However, this is not checked by the SIMATIC Manager and is thus the user's responsibility!

- Behavior of the objects of lower-level equipment modules:
  – A defined "Equipment module assignment" in the type is also copied.
  – An "Equipment module (basic requirement)" is not included in the copying because the basic requirement only abstractly describes the minimum requirements for the equipment module to be assigned.
  The corresponding equipment modules must be assigned at the instance for these basic requirements.
- Behavior of the objects of lower-level control modules:
  – Optional control modules are not copied.
  These can be selected and connected at the instance.
  Lower-level control modules can only be defined as "optional" on the first sub-level under the equipment phase.
  – A "Control module assignment" is copied as well, regardless of whether the assigned control module therein is a component of the type or a basic requirement.
  – A control module integrated as a component in the type is also copied when the instance is created.
  – A "Control module (basic requirement)" is not included in the copying because the basic requirement only abstractly describes the minimum requirements for the control module to be assigned.
  The corresponding control modules must be assigned at the instance for these basic requirements.
  – Commands and status can be defined at the types of control modules. These are not included in the copying.

The corresponding type is known at the instance in order to support the type-instance synchronization.

Comparison between type and instance

Synchronization of the type and the instances may be required when changes are made to the type or the associated instances.

You can find additional information on this in the section "Comparing and synchronizing the type and instances of an equipment phase (Page 329)".

11.3.3.9 Comparing and synchronizing the type and instances of an equipment phase

Introduction

The type and instance of an equipment phase are the same when the instance is created.

Synchronization of the type and the instances may be required when changes are made to the type or its associated instances.

Thus, for example, the sequencer of the equipment phase defined in an SFC type is identical when the instance is created. If changes are made to the SFC type in the master data library, however, synchronization has to be performed as usual by updating the SFC type in the project.

Comparison and synchronization

During synchronization, all instances of a type in the project are compared to the corresponding type in the master data library. This shows the differences between the types of equipment modules and their instances (process engineering level) and between the associated objects, such as CFCs (program level).

The synchronization of parameter descriptions and sequencers between the master data library and project is accomplished by updating the SFC type in the project. You can find additional information about this in the section "How to update block/SFC types in the multiproject (Page 110)".

Note

As of CFC V9.0, a VXM license as of V9.0 is required for type instance synchronization of technological objects, e.g. control modules (CMT).

You can find additional information and notes on synchronization in the section "Notes on type instance synchronization of technological objects (Page 347)".

Included objects

The following lower-level objects are included in the synchronization of the type and instances of an equipment phase on the plant level:

- Control modules
- Control variables:
  - Parameters
  - Signal
- Messages
- Commands and status
- Assignments of control modules
- Assignments of equipment modules
- Sequential control

Basic requirements for control modules and equipment modules, however, are generally not involved as they are by definition not permitted at the instances.

Note

Synchronization within the master data library

If a control module type is used in the master data library as a sub-control module at the type of an equipment phase, this sub-control module is also synchronized with its control module type.

Procedure

1. Select the project in the plant view.
2. Open the shortcut menu and select the menu command "Plant types > Synchronize...".
   The "Synchronize plant types" dialog box opens.
3. Select the desired types of equipment phases, equipment modules or control modules that you want to compare and synchronize in the left column of the table.
4. Click the "Synchronize..." button to start the comparison.
   Only relevant attributes and relations are included in the comparison.
   Functions and function blocks that have been added to an instance are not included in the comparison and are suppressed in the presentation of the comparison result.
5. The comparison result is displayed.
   The following objects are highlighted in the comparison result:
   – Objects added at the instance, for example sub-functions, messages, functions and function blocks, are marked as additional objects.
   – Deleted objects.
   – Objects with modified attributes.

   Differences in the topology of the sequencers and parameter descriptions are not shown directly in the comparison result. Rather, the change time stamps of the two compared objects are displayed:
   – The change time stamp of the SFC type in the master data library that is associated with the "Sequential control".
   – The change time stamp of the corresponding SFC type in the project.

   For a control module that is integrated in the type of the equipment phase, the corresponding "Control module assignment" is also shown in the comparison result. If the "Control module assignment" is missing or has been changed, you are prompted by an entry in the log file to check the control modules in the context of this equipment phase.
6. In the comparison result, select the instances to be synchronized to the corresponding type.
   Select or clear the check boxes next to objects at the left in the dialog.
7. Start the synchronization with the "Synchronize templates" icon.
   The selected instances are synchronized.
   Instance-specific extensions are retained if they do not apply to objects of the type.

Result

A comparison of the types of equipment phases, their lower-level objects and their instances has been performed and selected instances have been synchronized.

See also

Basics of the equipment phase (Page 299)

Creating an instance of the equipment phase type (Page 327)

11.4 The equipment module / equipment phase in data communication with COMOS

11.4.1 Overview of data exchange with COMOS

Introduction

The data generated on the engineering level in COMOS in the form of equipment modules, equipment phases and control modules can be imported into the automation layer of PCS 7. This forms the basis for creating sequential controls (SFC) and automation programs (CFC).

Overview of data exchange between PCS 7 and COMOS

The following figure shows the principle of data exchange between PCS 7 and COMOS using the automation interface (AI).

The following options are available for data exchange:

- Direct data exchange when COMOS and PCS 7 are installed on the same computer.
  This direct data exchange is described below.
- Indirect data exchange via an XML file when COMOS and PCS 7 are installed on different computers.
  You can find additional information on this in the section "Data exchange with COMOS via XML format (Page 336)".

Not only the types and instances of equipment modules are read from COMOS Integrated Engineering and PCS 7 and compared with the data exchange via the import service of the automation interface (AI). The plant hierarchy, types and instances of control modules, hardware configuration, etc. are compared as well. Changes can be deleted or applied to the target thereby.

Renamed objects

Renamed objects are recognized again during data exchange and displayed and compared when the data is synchronized.

The name change is detected for the following objects:

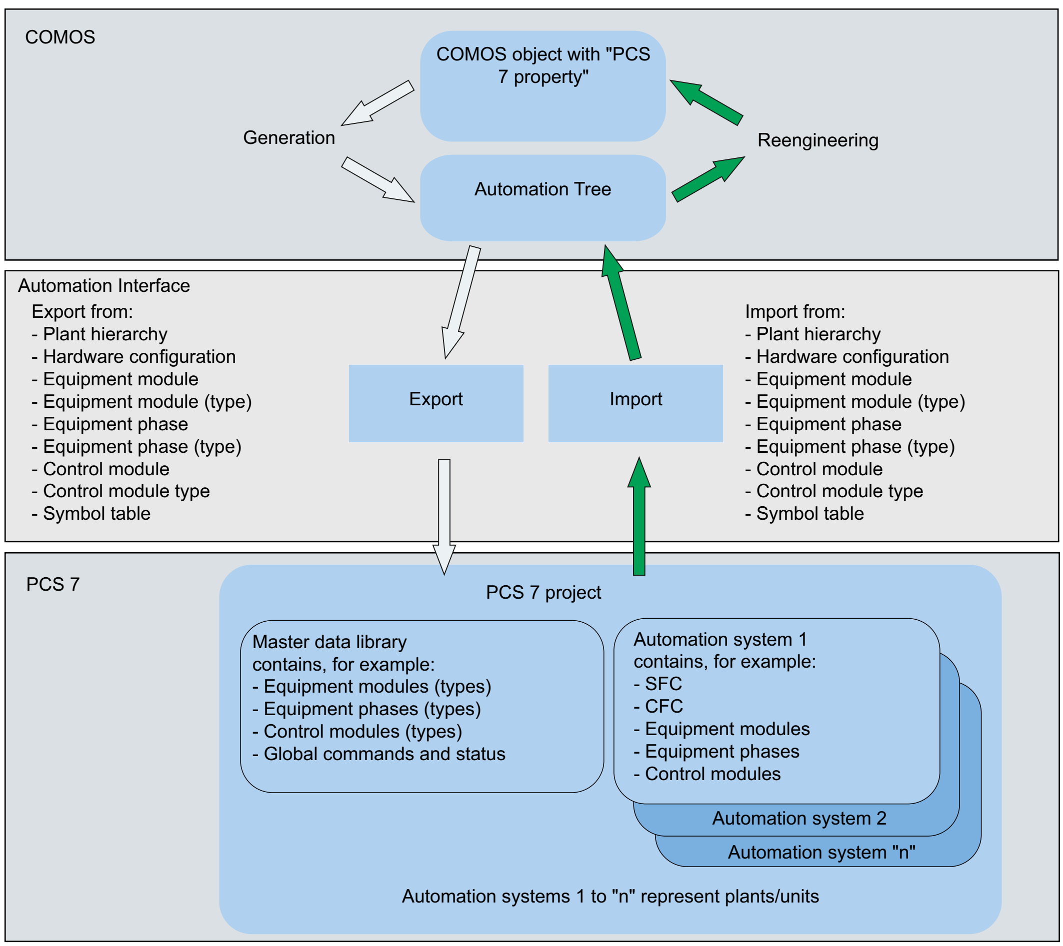
- Nodes of the plant hierarchy
- Equipment module (EM)
- Equipment phase (EPH)
- Control module (CM)
- Parameter, signal (CV)
- Parameters of a sequential control
- Sequencer of a sequential control
- Command
- Status

- Enumeration
- Enumeration value
- Hardware:
  - Station, slave (central, slave)
  - Subnet, bus (global subnet)
- Symbol (IO tag)

Graphic overview for data exchange

The data to be imported is read and compared in the Automation Interface (AI) of PCS 7.

All elements available for data exchange with PCS 7 are displayed in the COMOS Automation Tree.



Extended hardware parameters during data exchange

In addition to the hardware configuration, the extended hardware parameters are transferred during data exchange via the Automation Interface (AI). The data exchange of the extended hardware parameters takes place bidirectionally and can be enabled/disabled if, for example, not all extended hardware parameters are to participate in the data exchange for performance reasons.

- In COMOS, the transfer is enabled or disabled during import/export with the "Hardware > Extended parameters" option.
- In PCS 7, the transfer of the respective hardware parameters is enabled or disabled with an entry in the XML control file "AIaS7ExtHWParameter.xml".
  This file is stored in the directory "Program Files (x86)\SIEMENS\STEP7\S7DATA \AIS7AD".
  This XML file includes an entry for each hardware parameter that starts with the expression "<HWParameter name". This entry includes the expression "used" which is used to control the data exchange of this hardware parameter.
  - When the expression "used="1"", data exchange of this hardware parameter is enabled.
  - When the expression "used="0"", data exchange of this hardware parameter is disabled.

How data exchange between PCS 7 and COMOS works

1. Technical functions, which have been identified for the plant to be configured based on the customer requirements and the appropriate types of equipment modules or equipment phases, are specified, depending on the planned process sequences.
   The possible control strategies, setpoints and parameter descriptions (e.g. process values, parameters, timers, bit memory, control values, operator and note texts, position texts and messages) are defined for each type. The sequential control of the equipment module or equipment phase is described in the form of sequencers. An essential part of the type description for equipment modules is the assignment of the control modules (actuators and sensors), which are needed for the sequential control system.
2. The types of equipment modules or equipment phases are created with COMOS Integrated Engineering. Thereafter, these types must be transferred to the master data library ("Project Library") of PCS 7 to complement the implementation with the SFC and CFC editor there.
   Types and instances of equipment modules, equipment phases, control modules, the plant hierarchy, hardware configuration, etc. from COMOS Integrated Engineering and PCS 7 are read and compared in the data transfer dialog during data exchange. Changes can be deleted or applied to the target thereby.
   The types of equipment modules, equipment phases and control modules are provided in the master data library as templates for instantiation. When you import the instances, the types in the master data library are used as templates.

   Note

   Certain restrictions in PCS 7 have to be taken into consideration for a successful import from COMOS Integrated Engineering to PCS 7. You can find more information below.

3. After the types of equipment modules or equipment phases are defined and implemented, the instances can be configured. The instances are created and the connections are made to the specific control modules, representing the devices on the control module level of the unit.
   The instances with the assigned control modules are used as generation instructions on the process engineering level of COMOS Integrated Engineering in order to create, configure and interconnect the specific automation charts (CFC) in the corresponding PCS 7 project.

Relevant restrictions in PCS 7 for importing from COMOS Integrated Engineering

- Commands and status in the sequential controls:
  - Commands can only be used in steps in sequential controls.
  - Status can only be used in transitions.
- Instructions and conditions:
  - Step: A maximum of 50 instructions per step are possible.
  - Transition / Start condition: A maximum of 16 conditions are possible per transition / start condition, in groups of 2 x 5 and 2 x 3 conditions.

  You can find more information in the paragraph "Instructions and conditions of the steps and transitions" of the "The sequential control in the data exchange with COMOS (Page 339)" section.

See also

Basics of equipment modules (Page 264)

Basics of the equipment phase (Page 299)

11.4.2 Data exchange with COMOS via XML format

Introduction

Data can be transferred bi-directionally and compared with data exchange between PCS 7 and COMOS.

The following options are available for transfer and synchronization:

- Direct data exchange when COMOS and PCS 7 are installed on the same computer.
  The types and instances of equipment modules, equipment phases and control modules are read from COMOS Integrated Engineering as well as PCS 7 and compared via the import service of the automation interface (AI).
- Indirect data exchange via an XML file when COMOS and PCS 7 are installed on different computers.
  With indirect data exchange, data is first saved to an XML file using the export function in the SIMATIC Manager or COMOS Integrated Engineering.
  Then, the XML file generated by COMOS Integrated Engineering or PCS 7 is imported in turn into the SIMATIC Manager or COMOS Integrated Engineering. Synchronization is also performed with the import. Changes can be deleted or applied to the target thereby.

Indirect data exchange in the context of PCS 7 is described in this section.

Note

Security for data exchange via XML format

- Access to transfer media / transfer directory:
  When using the indirect data exchange between COMOS and SIMATIC PCS 7 via the XML format, make sure that only qualified personnel have access to the transfer medium or the transfer directory of the XML files.
- Digital signature of the XML file:
  When exchanging data via the XML format, the XML file can be digitally signed to describe the origin and integrity of the XML data. This corresponds to the requirements of the PCS 7 security concept (see "http://cache.automation.siemens.com/dnl/zI/zI1NDU1AAAA_22229786_HB/ps7sec_d.pdf").
  The verification of the digital signature ensures that the XML file on the transfer medium has not changed.
  The digital signature of the XML file is disabled by default. We recommend that you enable the digital signature.
  You can find more information on this in the "Requirements" section.
  Applied standards:
  The asymmetric signature process RSA according to the "PKCS#1" standard is applied (see "https://www.ietf.org/rfc/rfc3447.txt"). The attributes of the certificate are described under "http://www.ietf.org/rfc/rfc5280.txt".

Requirements

Software

An existing installation of the "SIMATIC XMLTRANSFER" software on the computer with the PCS 7 system is required for data exchange via XML file. The software is available on the Product DVD of COMOS.

Requirements for the digital signature of the XML file:

1. Enabling the digital signature:
   The signature of the XML file must be enabled in the PCS 7 project. The setting is made in the SIMATIC Manager using the menu command "Edit > XML data transfer > Settings for XML data transfer".
2. The following objects must be available on the computer in order to sign the XML file for export:
   - The certificate of the signing user.
     This certificate can be created by a commercial vendor or by a "Windows Server Certification Authority" and must be saved under "Current User > Certificates" in the "MMC" console of Windows. The corresponding so-called. "CA" certificate must be located in the "MMC" console below "Trusted Root Certificate Authorities".
     In this certificate, at least one of the values "digitalSignature" or "nonRepudiation" must be set in the properties of the "Key usage" attribute.
   - The so-called "Private Key" for this certificate.
3. The following objects must be available on the computer in order to import a signed XML file:
   - The certificate of the signing user. This certificate contains the so-called "Public Key", which is needed to identify the signature of the XML file.
     The certificate of the signing user must be saved under "Current User > Certificates" in the "MMC" console of Windows. The corresponding so-called. "CA" certificate must be located in the "MMC" console below "Trusted Root Certificate Authorities".

Data exchange of the extended hardware parameters

The data exchange of the extended hardware parameters depends on the settings in the control file "AIaS7ExtHWParameter.xml". You can find more information on this in the section "Overview of data exchange with COMOS (Page 332)".

Data transfer in the context of PCS 7

Bi-directional data exchange

Control module types can be specified and implemented in PCS 7 or they may already be available in a library. If these existing implemented control module types are to be used in COMOS, they must be transferred from PCS 7 to COMOS Integrated Engineering.

When changes are made on the engineering level or when types of control modules have been technologically specified in COMOS Integrated Engineering, these two kinds of types can be transferred from COMOS to PCS 7.

A type of equipment module or equipment phase is fully transferred with all control modules and optional control modules.

Default settings for data exchange via XML file

The following default settings can be configured in the "XML settings" dialog:

- Settings for the digital signature of the XML file
  You can find additional information about this in the "Requirements" section above.
- Filters for objects from the project and library

The "XML settings" dialog is opened in the SIMATIC Manager using the menu command "Edit > XML data transfer > Settings for XML data transfer".

Export and Import function in the SIMATIC Manager

The menu commands, "Export XML", "Export selected objects to XML" and "Import XML", are available for the export and import function in the SIMATIC Manager in the shortcut menu of the PCS 7 project.

These menu commands are only available if the "SIMATIC XMLTRANSFER" software is installed and can only be used for indirect data exchange with COMOS Integrated Engineering.

Import in PCS 7

- The "Import XML" import function mentioned above is used for indirect data exchange via an XML file. The user must provide and select the appropriate XML file for this.
  The import is started using the "Import XML" command from the shortcut menu of the PCS 7 project. As an alternative, you can start the process by using the "Edit" menu command in the main menu of the SIMATIC Manager.
- If the signature of the XML file is enabled, the certificate and the signature are verified during import. The conditions listed under "Requirements" must be met for this.
  - Upon successful verification, the data transfer dialog of the Automation Interface opens to display the comparison result and to start the import.
  - If the import of an XML file is performed without a signature although a signature is expected, the import process is aborted with a message.
  - If a certificate with the so-called "Common Name" and "Fingerprint" is found on the computer on which the XML file is to be imported and this certificate is also used in the signature of the XML file, a corresponding message is displayed.
  - If a change in the signature or the contents of the XML file is detected, a message is displayed and the user can decide whether the import is to be continued nevertheless.
- Once the import of a type of equipment module type or equipment phase into the PCS 7 project is successfully completed, in the import dialog the user can use the shortcut menu to directly open the equipment module or equipment phase in the master data library, where it is now available as a technological shell.
- The associated CFC only contains the instance of the SFC type as an equipment phase after the initial import.
  The user now has the possibility to implement the equipment module or equipment phase using the resources of the CFC and SFC editors.

Export from PCS 7

- The above-mentioned export functions are used for indirect data exchange via an XML file.
  - Total export using the menu command "Export XML"
  - Export of selected technological objects using the menu command "Export selected objects to XML"
    The menu command opens the dialog "Select technological objects" for selection of the objects

  The required export is started by using the "Edit > XML data transfer" menu command from the shortcut menu of the PCS 7 project.
  As an alternative, you can start the process by using the "Edit > XML data transfer" menu command in the main menu of the SIMATIC Manager.
  The user can select the path for the XML file.
- If the signature of the XML file is enabled, then a check is performed for the certificate to be used for signing the XML file for export. The conditions listed under "Requirements" must be met for this.
  - A message is displayed when the export and signature of the XML file are successful.
  - If the selected certificate is not suitable for the signing, the export process is canceled and a message appears.
- After that, the XML file is transferred and imported to the computer with COMOS Integrated Engineering.

11.4.3 The sequential control in the data exchange with COMOS

Introduction

The sequential control is also exchanged as part of the equipment module or equipment phase during data exchange with COMOS Integrated Engineering via the import service of the automation interface (AI). The data from COMOS Integrated Engineering and PCS 7 can also be read and compared. Changes can be deleted or applied to the target thereby.

The sequential control defines the sequencers and their parameter descriptions (e.g. setpoints, control strategies, parameters, messages).

The sequencers can contain the same elements as an SFC type:

- Steps
- Transitions
- Alternative branches
- Simultaneous branches
- Loops
- Jumps

The following describes how the equipment modules or equipment phases, sequencers and their elements participate in the data exchange.

Overview

Below is a description of how the components of the equipment module or equipment phase participate in the data exchange:

- Sequencer
- Steps and transitions
- Instructions and conditions of the steps and transitions
- Characteristics of the SFC type of the equipment module or equipment phase

The sequencer in the data transfer dialog box

In the data transfer dialog of the automation interface (AI), the sequential control and its elements are presented in a tree.

The elements of the sequencer are represented as sub-elements of the sequential control.

- Sequencer

  The sequencer always starts with a START step and ends with an END step. A start condition must always be be configured at the start of the sequencer.

  – Start condition

- A sequencer contains alternating step and transition elements.

  – Step

  – Transition

- In addition to a simple step and a transition, sequencers can also contain alternative and simultaneous branches.
  - Alternative branch

    An alternative branch contains at least 2 transition sequences.

    A transition sequence always begins and ends with a transition element. A transition sequence itself is a transition element.
  - Simultaneous branch

    An simultaneous branch contains at least 2 transition sequences.

    A step sequence always begins and ends with a step element.

  The step and transition elements are displayed according to the graphical and topological structure of the comparison tree in the automation interface.
- Sequencers can also contain loops and jumps. The loops and jumps are performed in accordance with a transition condition.
  - Loop

    A loop can be generally determined only by a step sequence. The transition condition is checked after the last step at the end of a sequence of steps and transitions. If necessary, the sequencer resumes at the first step of the sequence.
  - Jump

    A jump can cause the sequencer to resume at a given step depending on whether a transition condition is met.

Properties of the sequencer during data exchange and import

- A previously unavailable sequential control is fully transferred with the initial import operation.
- A sequencer in a sequential control can only be selected or deselected as a whole in the comparison tree of the automation interface.
  This means that only a complete sequencer can be included in an import or left out of it. This avoids inconsistencies.

Note

Existing steps and transitions during import into PCS 7

If a sequencer is marked and thereby included in the import, all the contents of the existing steps and transitions in the sequencer are overwritten in PCS 7. During import, corresponding messages are generated when objects are created, modified or deleted.

If changes were made to steps and transitions in the sequencers at the PCS 7 end after the import, these manual changes have to be checked, and possibly backed up before the import!

Order of sequencers in a sequential control

During the first import process from COMOS Integrated Engineering, the sequencers of a sequential control are sorted alphabetically.

This order does not make sense from the process engineering point of view. Therefore, the user can set the sequencers manually to the desired order in the SFC editor.

If the order was sorted manually and one or more new sequencers were added for the subsequent import, these are placed in front of the existing order of the sequencers. The new sequencers are thus shown at the far left in the graphical layout in the SFC editor.

Note

If the order of several sequencers in a sequential control was changed manually, the new order arising after new sequencers have been imported has to be checked by the user and modified, if necessary.

Steps and transitions in the data transfer dialog

1. Synchronization of the equipment phase first involves comparing the step and transition sequences with their steps, branches, transitions, etc.. Newly inserted, nameless topological elements, for example, simultaneous and alternative branches, loops, step and transition sequences can be correctly identified in this way.
2. Transitions and steps by definition have a name that cannot consist of numbers only. When steps and transitions are created in PCS 7, a name is not automatically generated or a number assigned for identification of these objects. Therefore, if there is no name at the PCS 7 end, a name is artificially created within the context of data exchange with COMOS Integrated Engineering. The name is formed by number. A "S" is prefixed for steps and a "T" for transitions.
   The user should assign suitable names for steps and transitions in the SFC editor of PCS 7 for the data exchange with COMOS Integrated Engineering.
3. Step and transition elements can only occur in accordance with the topographical rules of a sequencer; otherwise, the sequencer would be in an inconsistent state.
   If implausible sequences of steps and transitions are detected, an entry to this effect is made in the log file.

4. Steps and transition elements are compared in the comparison tree in the same way as all other objects. Therefore, only the objects involved are changed, added or deleted.
   Step and transition programs:
   The types of the equipment modules or equipment phases can be transferred with or without the step and transition program descriptions by COMOS Integrated Engineering.
   – Transfer without step and transition programs:
     The step and transition programs are not changed in PCS 7, even if the entire sequencer is selected for import as described above.
   – Transfer with step and transition programs:
     In the data transfer dialog, the step and transition programs are displayed with differences in the form of superfluous objects at the COMOS end. The objects of the step and transition programs can now be selected or removed for import.
     If an object is selected, it is also imported to PCS 7 and overwrites the existing contents.
     If an object is removed, no import is not carried out and the existing contents in PCS 7 are retained.
5. Step and transition elements cannot be selected or removed individually, but only indirectly via the selected or removed sequencer.
   Only the complete sequencer can be selected and thereby be included in the import.

Note

Meaning of the "Delete at target" option

If the sequencer is selected and thereby included in the import, steps and transitions are inserted or deleted at the target during import. This applies irrespective or whether the option "Delete at target" is activated or deactivated. Steps and transitions are also deleted in the target when the option "Delete at target" is deactivated. The decisive factor is whether or not the sequencer is included in the import process!

Existing steps and transitions during import into PCS 7

If a sequencer is marked and thereby included in the import, the existing steps and transitions in the sequencer are overwritten in PCS 7. You can find more information on this above in the section "Properties of the sequencer during data exchange and import".

Step and transition programs: If the types of the equipment modules or equipment phases were transferred with the step and transition program descriptions from COMOS Integrated Engineering, the user can specify whether these step and transition program descriptions are to be imported into PCS 7. You can find more information on under point 4.

Instructions and conditions of the steps and transitions

- The instructions of the steps and conditions of the transitions do not have own objects in the technological editor and are therefore not displayed for comparison in the data transfer dialog.

Note

Limits for instructions and conditions

The following limits in PCS 7 have to observed at steps and transitions for the transfer from COMOS:

- Step: A maximum of 50 instructions per step are possible.
- Transition / Start condition: A maximum of 16 conditions are possible per transition / start condition, in groups of 2 x 5 and 2 x 3 conditions.

If these limits are exceeded during the transfer from COMOS Integrated Engineering, the corresponding sequencer is not imported / generated in PCS 7 and a corresponding entry is made in the log file of the Automation Interface (AI).

Characteristics of the SFC type of the equipment module or equipment phase

During the first import process from COMOS Integrated Engineering, characteristics of the SFC type for the equipment module or equipment phase, such as control strategies, are also transferred in alphabetical order in PCS 7.

This order of the characteristics, particularly the control strategies, does not make sense from a process engineering point of view. Therefore, the user can manually arrange the characteristics in the desired order in the SFC editor.

Note

If the order of the characteristics of the SFC type of an equipment module or equipment phase was changed manually, the new order arising after new sequencers have been imported has to be checked by the user and modified if necessary.

11.4.4 Signal processing for data exchange with COMOS IE

Overview

Note the following in regard to the signal processing concept for data exchange with COMOS IE:

Control variables (of individual control modules, for example) of the "Signal" variable type that represent a channel request, should preferably be assigned to the corresponding block I/O of a technological processing block in the technological editor of the CFC instead of the I/O of the channel driver block.

This concept is only supported when using the PCS 7 APL library as of V8.0.

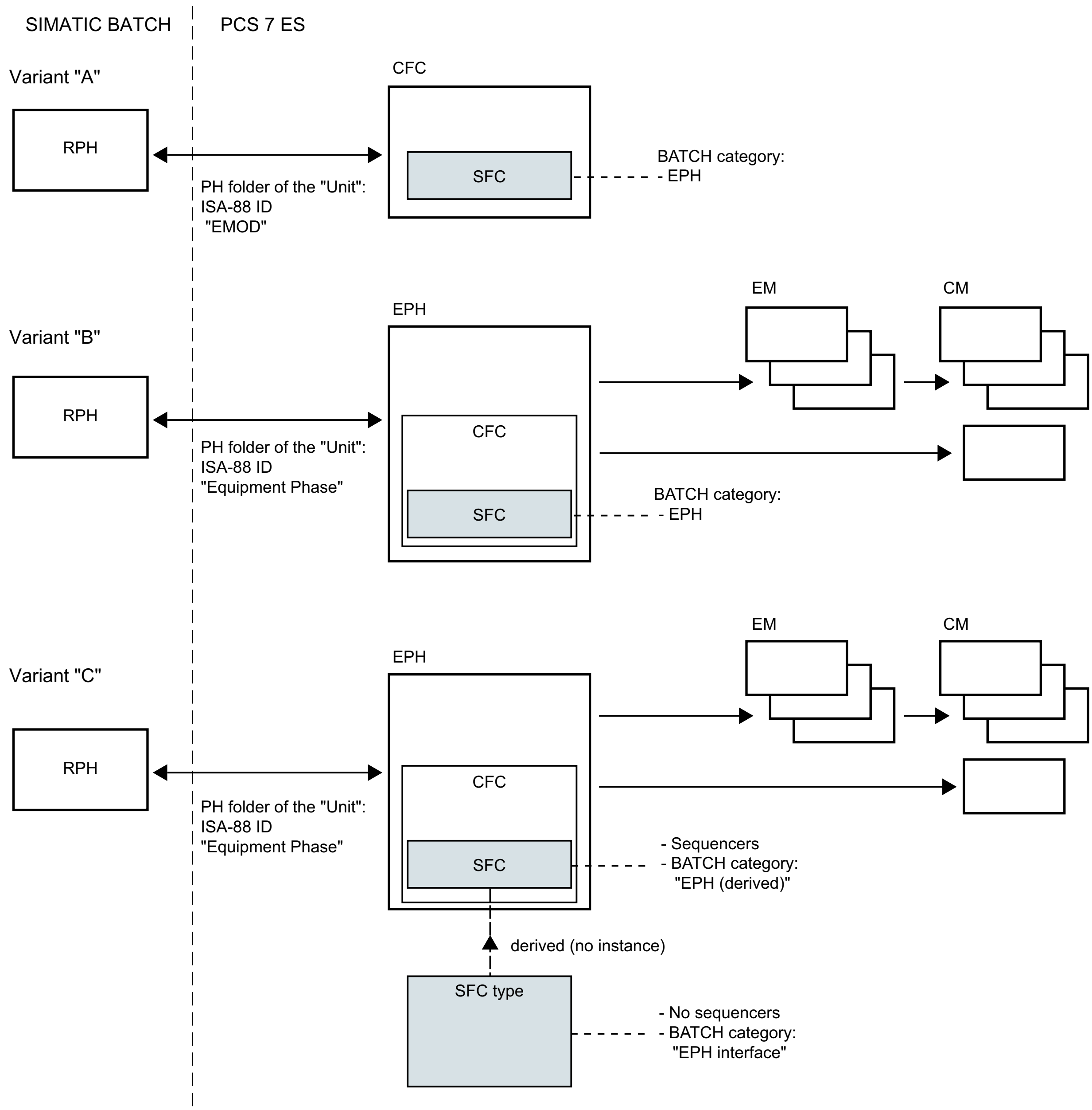
11.5 Configuration methods for assigning a recipe phase (RPH) to the engineering system

Overview

The following figure shows the possible configuration methods for assigning a recipe phase (RPH) to the engineering system.

- Variant "A"
  In the past, an assignment of the recipe phase (RPH) to an SFC block in the engineering system was used for connecting SIMATIC BATCH for process control in PCS 7. You can continue to use this method of configuration.
- Variant "B"
  With this configuration method, the equipment phase is the assignment partner for the recipe phase (RPH).
  Several lower-level equipment modules (EM) can be controlled with the equipment phase (EPH).
  Use:
  This configuration method is recommended if there is a process-related description at the process control end, in other words, the resources remain the same. The recipe creation in SIMATIC BATCH is independent of the resources.
- Variant "C"
  With this configuration method, an SFC block with the BATCH category "EPH (derived)" acts as an assignment partner for the recipe phase (RPH). It is derived from an abstract interface in the form of an SFC type with the BATCH category "EPH interface".
  Several lower-level equipment modules (EM) can be controlled with the equipment phase (EPH).
  Use:
  This configuration method is recommended if there are different process-related descriptions at the process control end, in other words, different resources are used. Therefore, the required setpoints, control strategies, etc. can be defined in abstract interfaces and then used in various "derivatives".
  The recipe creation in SIMATIC BATCH is independent of the resources.
  You can find additional information about configuring these variants in the section "Create an equipment phase with "derived interface" (Page 314)".

The equipment phase (EPH) and the configuration of the BATCH categories "EPH interface" and "EPH (derived)" are available as of PCS 7 V9.0.



See also

Basics of the equipment phase (Page 299)

11.6 Notes on type instance synchronization of technological objects

VXM license during type instance synchronization of technological objects is required

As of CFC V9.0, a VXM license as of V9.0 is required for type instance synchronization of technological objects, e.g. control modules (CMT), equipment modules (EMT), equipment phases (EPH).

The VXM license is required because this synchronization is based on VXM functionality.

If there is no VXM license available, a message is displayed that VXM has to be installed. However, the installation of the VXM license alone is sufficient to enable the VXM functionality for the type instance synchronization.

Type-instance synchronization of lower-level types/objects

If several types of technological objects are selected for synchronization during type-instance synchronization, and one of the selected types or objects is stored as instance below another selected type, only the higher-level type is synchronized and not the lower-level type or the object.

The following objects/types are not synchronized if they are nested in another type:

- Control module type (CMT)
- Command
- Status

An entry for the non-synchronized type is made in the synchronization log, for example:

"The object <Path to Object1> is not synchronized because the type <Name of Type2> is itself selected for the synchronization."

Example of log entry:

- Configuration:
  "Object1" is a control module type (CMT), which is stored as instance in the EM type <Name of Type2>.
- Result:
  - The EM type <Name of Type2> is synchronized with its instances.
  - The instance of the CM type "Object1" in the EM type is not synchronized.

Remedying of the cause for the log entry:

1. After the synchronization, check the log for entries in this format:
   "The object <Path to Object1> is not synchronized because the type <Name of Type2> is itself selected for the synchronization."
2. Repeat the synchronization process and select only the types/objects that are named as "Object1" in the log entry.
3. Repeat the synchronization process and select only the types that are named as "Name of Type2" in the log entry.

11.7 Configuring a global command or status

Introduction

The "Command" and "Status" objects can be defined as follows:

- At the type of a control module (not at an instance) or at a "Control module (basic requirement)".
  These commands and statuses are specific to these objects.
- Globally, i.e. with the SIMATIC Manager in the master data library.
  These commands and statuses are generally based on equipment phases (EPH) or equipment modules (EM). They can be used in the "Sequential controls" (SFC type) of the equipment phases or equipment modules.

The following section describes how a global command or status is created. The optional creation of formal parameters for these objects is also described.

Note

Global commands and status in the library

As of PCS 7 V9.0, pre-defined global commands and statuses are available in the "SFC Library". These can be copied and used in the "Sequentical control" of equipment phases or equipment modules.

Distinction by mode of action

Global commands and statuses are distinguished according to their mode of action:

- Commands/statuses acting on local states of the SFC instance.
  This means that the implementation only uses operating states, sequencer information, etc. of a sequencer.
- Commands/statuses acting on lower-level sequential controls.
  These relate indirectly to setpoints, states, etc. of an SFC instance of a lower-level equipment module. An "Equipment module assignment" is available for this in the higher-level equipment phase or equipment module.
  The "Access to lower-level equipment module" attribute is enabled for these commands/ statuses.

General information

Commands and states are identified by a name and there can be more than one. These objects can be selected, copied and deleted individually or collectively in the SIMATIC Manager.

Naming global commands and status

The maximum length of the name of a global command or status is 22 characters because only these 22 characters will be accepted during subsequent configuration of the assignment in the technological editor.

Note

Use of characteristics in the conditions of a global command/status

No tags that can be used directly in the conditions exist for the following characteristics:

- Control strategy
- Note text
- Position text

When the standard parameters "QCS", "QCSP", "OPTINO" and "POSINO" are used in a condition of a global command/status, these above-mentioned characteristics can only be addressed via the number, for example of the control strategy.

For this purpose, the following configuration in the global command/status is required:

- To use one of the above-mentioned characteristics in a condition, a corresponding formal parameter must be created as lower-level object of the global command/status for this characteristic. This formal parameter must have the relevant data type.
  Example: A position text has the name "Positiontext_A" and the data type "POSITION TEXT".
- In the assignment of the condition, the name of this formal parameter, e.g. "Positiontext_A", is entered and supplemented by the string "_NUMBER". The field automatically gets a yellow background.
  In this example, the condition would be called: "POSINO" = "Positiontext_A_NUMBER".

When data is taken from COMOS Integrated Engineering or the runtime program is generated in PCS 7, the corresponding number of the characteristic, for example, of the position text, is automatically inserted into the condition in the sequencer of the equipment phase (EPH) or equipment module (EM).

In this example, the condition would be called: "POSINO" = "1".

Requirement

A multiproject open containing a master data library is open in the SIMATIC Manager.

Procedure

1. Use the menu command "View > Plant View" to open the plant view.
2. In the plant view, navigate in the master data library to the directory where you want to create the global command or status, for example "Process tag types". Other directories can be used, for example, ones you have created in the master data library.

3. Select the command "Insert New Object" and then "Command" or "Status" in the shortcut menu.
   The newly created command or status appears in the right window with the corresponding icon.
   – Status
     
     A lower-level "Parameter" object named "OUT" is created automatically for a new status. This parameter has the BOOL data type and serves as the formal output parameter.
   – Command
     
4. Open the command or status in the CFC editor to configure its properties.

   Note

   As of CFC V9.0 SP4, you cannot insert objects (such as blocks, sub-charts, etc.) from catalog or by copying from other charts to the CFC chart of global command or status (that is, if the command or status is the root-level control module). Such commands and status can be edited only in the technological editor. However, you can edit the lower-level command or status in both technological editor and CFC chart.

   Select the icon for the "Command" or "Status" in the technological editor.
   The command or status has the default attributes "Name", "Author" and "Comment".
   Configure the desired attributes.
5. Optionally, you can create lower-level formal parameters for this command or status.
   To do this, select the icon for which you want to create a parameter in the technological editor.
6. Select "Insert New Object > Parameter" from the shortcut menu.
   The selected object is inserted and displayed as an icon.

7. If you want to configure the attributes of the parameter, select the "Parameter" icon in the technological editor.
   The associated attributes are displayed.
   Configure the desired attributes.
8. To configure the properties of the new command or status, select the corresponding icon in the technological editor.
   In the shortcut menu, select the menu command "Properties".
   The "Properties" dialog box opens.
   For "Command", the dialog has a similar structure to the dialog for step configuration; for "Status", the dialog has a similar structure to the dialog for the configuration of transition conditions.
9. Configure the desired conditions for the command or status.
   You can also use the "Browse" button to open a selection dialog. All the parameters that can be used, which means only the block I/Os that have an identifier "S7_contact = true", are displayed in this dialog box.

10. Select the desired parameter. Click the "Apply" button to apply the selected parameter in the line previously selected of the "Properties" dialog box.
    In the "Properties" dialog box, the next row of conditions is selected automatically and the selection dialog box remains open. If needed, repeat this step to configure additional conditions.
    Click "Close" to close the selection dialog.
11. Click "Close" to close the "Properties" dialog box.
    If required, a prompt for applying the configuration is displayed. Confirm this prompt.

Result

The global commands and/or states have been created in the master data library.

Formal parameters for these objects have been optionally inserted.

A parameter named "OUT" is created automatically at the status.

See also

Overview of data objects of the equipment module and control module (Page 269)

Overview of the data objects of the equipment phase (Page 304)

Overview of the data objects of global "Command" and "Status" (Page 352)

11.8 Overview of the data objects of global "Command" and "Status"

Overview

The "Command" and "Status" objects can be defined as follows:

- At the type of a control module (not at an instance) or at a "Control module (basic requirement)". These commands and statuses are specific to these objects.
  You can find a description of this in the tables in section "Overview of data objects of the equipment module and control module (Page 269) ".
- "Globally", i.e. with the SIMATIC Manager in the master data library.
  These commands and statuses are generally based on equipment phases (EPH) or "Sequential controls" of equipment modules (EM) and can be used in all equipment phases or "sequential controls" of a plant.

These global commands and statuses are described in the table below.

You can find more information about configuration and about characteristics in conditions in the section "Configuring a global command or status (Page 348)".

Object/icon	Description
Commands 	• "Global" commands are generally based on equipment phases (EPH) or "Sequential controls" of equipment modules (EM) and can be used in all equipment phases or "sequential controls" of a plant. This means that the implementation only uses operating statuses, sequencer information, characteristics, etc. of a sequence control. • A command is determined by its name. You can find additional information about naming in the section "Configuring a global command or status (Page 348)". • Global commands share a namespace in the master data library with the following objects: – Types of equipment module and control module – SFC and CFC – Global status • Each command has the attributes "Comment" and "Author". • A command can be provided with one or more formal parameters. • There can be more than one command. They can be selected and copied individually or collectively in the SIMATIC Manager.
Statuses 	• "Global" statuses are generally based on equipment phases (EPH) or "Sequential controls" of equipment modules (EM) and can be used in all equipment phases or "sequential controls" of a plant. This means that the implementation only uses operating statuses, sequencer information, characteristics, etc. of a sequence control. • A status is determined by specifying its name. You can find additional information about naming in the section "Configuring a global command or status (Page 348)". Global statuses share a namespace in the master data library with the following objects: – Types of equipment module and control module – SFC and CFC – Global commands • Each status has the attributes "Comment" and "Author". • A status can be provided with one or more formal parameters. • There can be several states. They can be selected and copied individually or collectively in the SIMATIC Manager.

Compiling 12

12.1 Essentials for CFC chart compilation

General

If you want to control the assignment of resources on the target system via the "Settings for compilation" dialog box, it is important to understand the relationships explained below.

You can find additional information in the section: Settings for compilation/download (Page 356)

The structure you configure in CFC is mapped to the S7 objects FC and DB.

FCs are required to call the blocks processed in the CFC chart according to the configured run sequence, as follows:

- One FC for each OB used
- One FC for each runtime group

DBs are used as follows:

- An instance DB is created for each instance of an FB.
- Internal DBs are created in CFC for storing intermediate results, for example, from FCs. One of these DBs is required for each data type. An additional DB will be created when the maximum length of 4 KB is reached.

SCL compiler

Note the following when compiling with the SCL compiler:

- The time required by the SFC compiler may be extended considerably due to large CFC structures. In some cases, there may be no progress visible for several minutes.

12.2 Settings for compilation/download

Making the settings

You can customize settings for the current CPU prior to compilation.

Select the menu command **Options > Customize > Compile/Download…**.

A dialog box opens. in which you can:

- Set warning limits in order to detect possible risks in due time before you start your download. Warning limits are available for:
  - Local Data
  - Instance DB of the S7 communication (number)
  - Load memory/work memory
    The check is only performed when the download is carried out.
  - Blocks per runtime group or OB

  You can find additional information for these terms in the help "Help on Compiling and Downloading S7 Programs".
- Specify which resources should remain unused when compiling the charts of the current chart folder (areas reserved for other applications). This may be useful, for example, if you want to solve an automation task partly with charts and partly by programming, for example, STL, LAD or SCL programs, and when you are using functions (FCs) or data blocks (DBs) from other sources in your user program.
- View the statistics of resources (DBs, FCs) available on your CPU for compiling the charts and how many of those are currently being used. The following information is provided:
  - The maximum available DB and FC numbers on the CPU
  - The DB and FC numbers available for CFC (maximum CPU minus the reserved areas)
  - The number of DB and FC numbers in use by CFC

  If no CPU has been assigned to the active chart, the "Maximum" and "Available" columns have the entry "- -".
- Check the relevant box to generate an image of the downloaded program for comparison purposes:
  With this option, you can generate an image of the current program following a successful download operation. The image is created as an XML document and assigned to the program.
  Then, when you want to download again, you can click "Show Changes" in the "S7 Download" dialog box to run a comparison between the XML file and the data you want to download before it is actually downloaded. You can only carry out this comparison if you have the Version Cross Manager (VXM) add-on package installed and if an image was generated when the program was previously downloaded.
  Please note that XML file generation slows down a full download.

Compressing

Click "Compress" to call a function you can use to close gaps in the DB and FC number pool. Such gaps develop, for example, when you delete objects. Compressing eliminates gaps in the pool DB. DB numbers may then be free again.

Note

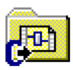
If you have changed and compressed the number ranges, you must compile the entire program and download it to the CPU in stop mode.

12.3 Compile charts as program

Overview

When objects have been added or modified in a chart folder but not yet compiled, this is indicated at the icons in the SIMATIC Manager:

- By an additional "C->" symbol in the bottom left corner on the closed chart folder. This means that the program still needs to be compiled.

- By an additional "C->" symbol in the bottom left corner at the icons of objects in the chart folder. This means that this object has been changed but not yet compiled and downloaded. Example with an icon of a CFC:

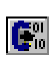
You can now compile the entire program or all changes as described below.

If you do not want to compile the entire user program or all changes, the "Selective Download" function is available for compiling individual charts and downloading them to the CPU. You can find additional information on this in the section "Selective download of individual charts (Page 378)".

Compiling

A consistency check is performed automatically during compiling. You can also start this consistency check without compiling, for example, before you overwrite an old and still functioning CFC program by compiling. To do this, select the menu command **Chart > Check Consistency > Charts as Program.../Chart as Block Type...**.

To compile, select the menu command **Chart > Compile > Charts as Program...** or click the icon in the toolbar:



A dialog box opens. You can select other options here.

Logs are generated during the consistency check and compiling. You can view the entries in CFC with the menu command **Options > Logs...**.

The consistency check differs, depending on the specific target system. The following checks are performed, for example, to see:

- Whether the in/out parameters or block outputs of the type "ANY", "STRING", "DATE_AND_TIME" or "POINTER" were interconnected
- Whether blocks are installed according to their task list
- Whether FC and DB numbers are in the range that is illegal for CFC (areas reserved for other applications)

Scope of compiling

You can select one of two compiling options:

- **Scope: Entire program**
  When you set this option, all AS resources will be compiled, in other words the full content of the chart folder, regardless of any changes.
- **Scope: Changes only**
  When you set this option, you only compile the changes made in AS resources since the last compilation. This only includes changes in content, for example, interconnections, block attributes or number of inputs, but not the movement of objects within the chart. Select "Compile changes" wherever possible, since it significantly reduces compilation time.

Note

When you compile changes, only the scope of your changes will be checked. You need to perform a consistency check to verify the entire program.

Optional functions for execution prior to compilation

Before you compile your data, you can set specific optional functions to be executed prior to compilation:

- **"Generate SCL source" option**
  The check box is disabled as the default setting. If this box is checked, an SCL source file is generated and stored in the source files folder. This source is not needed for compiling the program.
  Generating the source is only relevant to you if you want to view certain program sections in SCL code, for example, to make things clearer or for use in troubleshooting.
  The enabled check box is only valid for one compile; at the next compile it is disabled again.
- **"Generate module drivers" option**
  The check box is selected by default, in other words, the driver generator will also be started prior to each compile. In special cases, such as incomplete hardware, you can clear the check box so that the "Generate module drivers" function is not executed.
  This deactivation only applies for this compile, for the next compile the check box is reactivated.
- **"Module drivers settings" button**
  The driver generator creates and interconnects module drivers for the existing signal-processing blocks, if the hardware configuration has changed since the last compilation run.
  If you do not want to use the driver blocks from the current PCS 7 Library, you can use the "Module driver settings" button to open a dialog box where you can select the required driver library.

Note

Changes to sample times in HW Config take effect the next time the ES is compiled.

After compiling

When the compilation is completed or canceled, the S7 Logs dialog box opens. All compilation events, warnings and errors are included in this log. Based on this log you can verify compilation and print the log file by clicking "Print". Click "Close" to close this dialog box.

You can open and print the log subsequently with the menu command **Options > Logs...**.

Note

Please observe the following information:

- The warnings written to the log did not abort compilation, but could cause errors during subsequent downloads, in other words, prevent or cancel the download.
  Warnings relating to textual interconnections that are not closed or interconnections to addresses not (yet) available in the symbol table do not affect the download. In both situations, a substitute value will be generated according to the default parameter value of the block type.
- A warning is written to the log if your project contains blocks that are called recursively. Recursive blocks are displayed in the "Block Call Hierarchy" view in the chart reference data. You can call this view using the menu command **Options > Chart Reference Data**.
- Since PCS 7 blocks do not contain recursive calls, these must be user blocks. Make sure that any abort conditions defined in your program are always met so that no endless loops can occur, or remove the recursion.
- If the license is not adequate you can only compile the entire program. Compiling of changes is canceled in this case.
- Compiling the entire program does not necessarily mean that a complete download is necessary. If the program was already loaded on the CPU prior to compiling, it is possible to perform a download of changes.

Additional information

You can find additional information on this in the following sections:

Essentials for CFC chart compilation (Page 355)

Settings for compilation/download (Page 356)

Compile and download objects (Page 377)

12.4 Compile chart as block type

Start

You can create block types from existing CFC charts that will be used more than once and that have the required chart I/Os. You can assign system attributes to these block types.

To perform the compilation, select the menu command **Chart > Compile > Chart as Block Type....** A dialog box opens displaying the “General” and “Attributes” tabs.

Compiling

You have the following compilation setting options:

- In the "General" tab, specify the block type properties before starting compilation (FB number, symbolic name, name (header), family, author, version) and specify the target system on which the block will be used. This information (S7-300/S7-400) is relevant for the startup blocks, since the SCL compiler needs to generate a different code for each target system.
  Note on the FB number: This number must not exceed the range supported by your destination CPU.
  Example: In the case of 414-2DP CPUs, the maximum number range is set to 512 (address range: number of FBs). The number you assign to the block must, therefore, not be higher than 511.
  The use of specific code sequences ensures that all chart blocks to be compiled are called according to their entry in the S7_tasklist attribute.

  Note

  An S7 -300 supports only OB100, an entry of any other OB in the task list is not permitted. Otherwise an error message results.

- You can start a code optimization that affects the local data requirements or the online download of changes.
  - Local data requirements:
    With this type of optimization, a change in the chart does not increase the local data requirements, since all temporary variables are stored in the instance DB (VAR area). This does, however, lead to a change in the structure of the instance DB and to a change in its interface time stamp. In this case, no online download of changes is possible.
  - Downloading changes in RUN:
    With this type of optimization, if there is a change in the chart, the temporary variables will be stored in the VAR_TEMP area if possible. Possible in this context means that all intermediate results in the data flow are stored at this location. Only the intermediate results that do not exist in the data flow, for example, in feedback loops, continue to be stored in the VAR area (instance DB). The advantage of this optimization method is that not all modifications will affect the interface time stamp of the instance DB, in other words, in most cases an online download of changes will still be possible. However, the local data requirements are increased.
- As an option, you can also enable know-how protection. The algorithm of the block can be viewed, but only modified if the suitable SCL source files exist.
- You can set the system attributes for this block type in the "Attributes" tab.

Consistency check

Consistency is checked against the following criteria during compilation:

- Whether the chart contains nested charts
- Whether the block is installed manually only once in an OB, in other words, only once outside the OB specified by the "S7_tasklist" system attribute
- Whether all blocks are installed consecutively in the same OB, so that no blocks of other charts are placed in between
- Whether the blocks are inserted into runtime groups
- Whether all chart I/Os are connected to internal block I/Os
- Whether the chart I/Os have an EN input, since these are generated automatically
- Whether an existing ENO I/O is assigned the BOOL data type
- Whether block inputs that cannot be interconnected are assigned the attribute S7_link:="false" at the chart I/O
- Whether illegal blocks exist
- Whether illegal data types exist at the I/Os

Additional information

- In the SIMATIC Manager in the online help "Help on System Attributes" in the sections "System Attributes for Blocks", "System Attributes for Block I/Os" and "Assigning System Attributes".
- Creating block types in CFC (Page 104)

12.5 Special considerations during compilation for a CPU 410-5H PA

Overview

When compiling the program for a CPU 410-5H Process Automation (CPU 410-5H PA), the following special considerations apply:

- Monitoring block for license information
  During compilation, a "@PA-CPU" CFC is automatically created which includes an instance of the "PA_CPU" block. This CFC is also installed in the OB1.
  The "PA_CPU" block is required in the CPU 410-5H PA and is used for license monitoring. It is available as FB16 in the directory "ELEM_400" in the CFC library.

Downloading 13

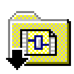
13.1 How to download a user program to the target system

Overview

After you have completed compilation, you can download the user program and test and run it on the target system.

When a program or chart has been compiled but not yet downloaded to the CPU, this is indicated at the icons in the SIMATIC Manager:

- By an additional arrow symbol in the bottom left corner on the closed chart folder. This means that the program still needs to be downloaded after being compiled.

- By an additional arrow symbol in the bottom left corner on the icons of objects in the chart folder. This means that this object has been compiled and not yet downloaded.
  Example with an icon of a CFC:

You can now download the entire program or any changes as described below.

If you do not want to download the entire user program or all changes, the "Selective Download" function is available for downloading one or more charts. You can find additional information on this in the section "Selective download of individual charts (Page 378)".

General information

Please observe the following information:

- Always download programs created in CFC to the target system from CFC, since only the download function of CFC ensures consistency of offline and online target system data.
  Exception:
  The CFC download function is also offered in the SIMATIC Manager in the following cases:
  - For a selected chart folder with the menu command **CPU > Download**.
  - For a selected (multi)project or a station with the menu command **CPU > Compile and Download Objects**.
    You can find information on this in the section: Compile and download objects (Page 377)

  What is not allowed is marking the block folder or individual blocks and calling the menu command **CPU > Download** or copying the blocks from the "Offline block folder" and then inserting them into the "Online block folder".
- Downloading to the CPU, just like working in test mode, is a protected function in S7 that must be logged if the SIMATIC Logon Service is installed. The actions during downloading and the time stamps are listed in the ES log if the current chart folder for the ES log was enabled with the menu command **Chart Folder > Object Properties... > "ES Log"** tab.
  You can find additional information on this in the following section: Change log and ES log (Page 401)
- If warnings are entered in the log during compilation of the program, this may mean that downloading will be prevented or aborted. In this case, check whether warnings in the compilation log are relevant to downloading before you download.

Procedure

To download the user program to the target system, proceed as follows:

1. Select the CFC menu command **CPU > Download...**
   or
   click the following button in the toolbar:
   

   A dialog box opens where you can select the type of download.
   If changes to data relevant for the download have been made in the user program prior to downloading, a message appears, which informs you that the program first needs to be compiled. You are then asked whether you want to perform that compilation, followed by the download.
2. Select the type of download:
   – Entire program: All applicable program objects will be downloaded.
   – Changes only: Changes that have been made since the last download are downloaded.
   – Download to test CPU (entire program): All applicable program objects will be downloaded.
   – Download changes to CPU in RUN mode:
   You can find more information on this in the description below.
3. Click "OK".
   The download begins.

Click "Apply" to save your changes without initiating the download.

If you only want to save your settings without downloading them, click "Apply" and then "Cancel".

Download: Entire program

The "entire program" can be downloaded in the STOP or RUN-P modes. In RUN-P mode, the CPU is set to STOP following a prompt and all the blocks it contains are deleted. After the download has been completed, you are prompted "Do you want to restart the CPU?". Click "Yes" to trigger a CPU restart.

Before a complete download is performed, a dialog box opens and you have the option of reading back the parameters. Readback is only possible if no interface changes have been made to block types, which means a download of changes remains possible.

The following applies when performing a complete download:

- Compiling the entire program does **not** necessarily mean that a complete download is necessary. If the program was already loaded on the CPU prior to compiling, it is possible to perform a download of changes.
- If a complete download has been canceled, it is no longer possible to download changes until a complete download has been fully executed.
  Reason: The blocks were deleted on the CPU prior to the download.

Download: Changes

You can download changes when the CPU is in RUN-P mode. The correct download order and segmentation required for the download of data will be maintained. Please note that there is no absolute guarantee that the CPU will not change to STOP as a result of temporary inconsistencies, since not all conditions can be checked.

If the "Include user data blocks" check box is selected (default setting), the following conditions apply to data blocks located outside of the CFC area when downloading changes:

- The blocks are included in the download if the time stamps differ or data blocks have been added.
  Changed values in the CPU are always overwritten by the content of the DB in the offline program.
- The blocks are deleted on the CPU if they do not exist in the S7 program.

If user data blocks are to be ignored when downloading changes, deselect the check box.

This can have the following consequences:

- If a user data block exists in the S7 program but does not exist on the CPU, downloading will be aborted with an error message.
- If the user data block exists on the CPU but no longer exists in the S7 program, a warning is displayed. A warning also appears if the user data block in the S7 program differs from that on the CPU. Correct execution of the program is then the user's responsibility.

Notes on downloading changes

Please observe the following information:

- It is always possible to download changes if the following conditions are met:
  - When a complete download has been successfully performed at least once
  - The download is performed with the program structure previously used.
    You can find information in the section: What you should know about downloading changes (Page 372).
- Each time you download changes, the "asynchronous compression of CPU memory" function is executed implicitly. This may help to prevent warnings and error messages during a later download of changes due to lack of memory.
  Compressing the CPU memory has other effects on downloading compared with compressing the DB/FC number areas. In the latter case, no download of changes is possible.
- Compiling the entire program does not necessarily mean that a complete download is necessary. It is possible to download changes if the program was loaded on the CPU prior to compilation.
- You can compile a program as often as you wish (changes only or complete program) without losing the ability to download changes.
- You can repeat a download of changes you have canceled in order to download objects not included the first time.

- You cannot download changes after you have tested a modified program on another system or via PLCSIM before downloading it to the runtime system.
  Remedy: With the option "Scope: Download to test CPU (entire program)", it is still possible to download changes to the original CPU. See also:
  Downloading a modified program to a test CPU (Page 371)
- Answers to the questions:
  – "What situations will prevent online downloads of changes?"
    and
  – "How do I maintain my program's ability to perform online downloads of changes?"
    can be found under: What you should know about downloading changes (Page 372)

    Note

    Certain scenarios may mean that a download of changes can no longer be performed. The only option is to download the entire program with the CPU in STOP. Before this is carried out, a warning appears to guard against unintentional loss of the capability to download changes.

Download: Download changes to CPU in RUN mode

This option is only available with a CPU 410-5H PA, because this CPU supports type update in RUN mode. The operating modes for the various download functions differ with the CPU 410-5H PA from those of other CPUs.

You can find additional information on this in the section "Special considerations during download to a CPU 410-5H PA (Page 386)".

System support

Reference lists will be created for blocks compiled under STEP 7 => V5.0 + SP3 or SCL => V5.0 + SP3. Based on those reference lists, the system performs a check in order to detect causes of CPU STOP and to prevent downloads if any errors (messages) have occurred.

You can find additional information on this in the following section: System support for avoiding causes of CPU STOP (Page 374)

Displaying changes before download

The "Show Changes" function is only available if the Version Cross Manager(VXM) add-on package is installed and if an image of the downloaded program was generated.

Generating an image of the downloaded program

Assuming you have selected "Generate image of downloaded program for comparison" in the "Settings for Compilation/Download" dialog box, the image will be generated as an XML document and assigned to the program following a successful download operation.

Comparing programs

If you generated an image of the downloaded program, you can click "Show Changes" in the "S7 Download" dialog box to run a comparison between the XML file and the program you now want to download before it is actually downloaded.

VXMis called for the comparison. The comparison will enable you to see which data have changed in relation to the program that was downloaded previously. You can then decide whether or not to download the latest version.

Notes on H CPUs

- If the H CPU is being operated in solo mode, for example, due to CPU failure, and a CPU failover has taken place, a dialog will appear following an online access request. There you can select which CPU should be linked. This dialog box does not appear in redundancy mode.
- All changes will be lost if you download program changes to a CPU operating in solo mode and then execute "Switchover with modified configuration" with the menu command **CPU > Operating Mode...**. In this case you need to download the entire program.
  Remedy: Download in redundant operation. In this case, you must make sure that the operating mode remains unchanged until the download is complete.

Note on F systems

You must enter the F password in order to download changes from programs with a modified F component. Otherwise, the F system will reject the download.

13.2 Downloading a modified program to a test CPU

Downloading to a test CPU

With the option "Download to test CPU" in the S7 download dialog, you can download a changed program for testing to a different CPU or to PLCSIM. The ability to download changes is not lost.

When you download as a test, the entire program is downloaded to the test CPU or to PLCSIM without losing the download identifiers and without the comparison stamp being written to the ES data management.

To avoid accidental overwriting of the program by a test download, the loader checks whether the changed program is capable of a changes-only download. If this is the case, the original CPU must be involved and not the test CPU. A note text to this effect is then displayed and the download cannot be performed. The “OK” button is disabled.

Additional information

You can find additional information in the section: What you should know about downloading changes (Page 372)

13.3 What you should know about downloading changes

How do I maintain my program's ability to perform online downloading of changes?

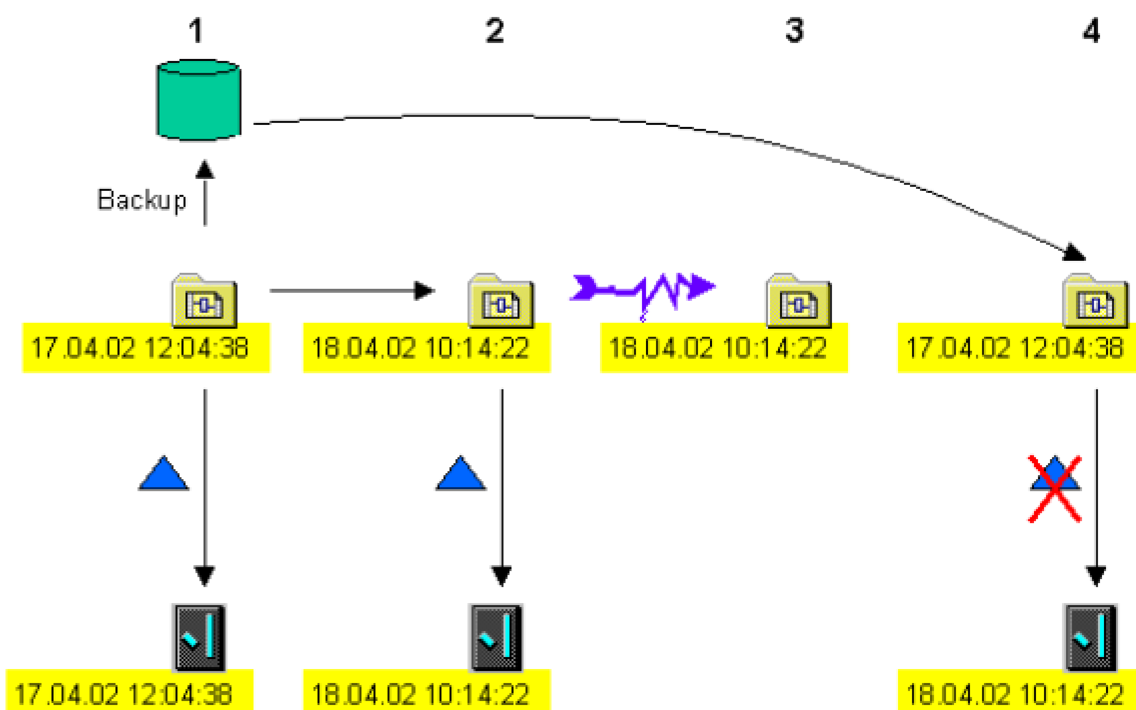
In cases where plants change during operation but must not be set to STOP mode under any circumstances, the possibility of downloading changes to them must be ensured.

Note

If you want to ensure that your program retains its ability to download changes, you should generate a backup copy after each completed download.

This measure may not be required for the testing or commissioning phase. The decision is yours and depends whether or not a complete download in STOP can be tolerated under specific circumstances.

The figure below shows how the online ability to download changes can be lost unintentionally.



Legend:
Blue triangle = download of changes

Step 1:
The program was downloaded to the CPU and a backup copy was generated. At the time of download the time stamp generated during compilation is stored in the offline and online

program (17.04.02 12:04:38).
Therefore, two time stamps exist in the offline program:

- One which is updated each time a compilation is performed
- One which is used for comparison and is only overwritten once the download has been successfully completed

Step 2:
The offline program is assigned the current time stamp "18.04.02 10:14:22" after changes in the program and subsequent compiling. The offline comparison stamp generated during the last download points to "17.04.02 12:04:38"; the online program also contains this time stamp. A download of changes is carried out. The offline program is assigned the new comparison stamp "18.04.02 10:14:22" that will also be transferred to the online program.

Step 3:
The program is changed and becomes inconsistent, for example, after data is lost due to a power failure.

Step 4:
The system will refer to the last known good version (step 1) and execute the changes made in steps 2 and 3 once again. The program is assigned the current time stamp "20.04.02 09:05:35" after it has been compiled, the offline comparison stamp of the last download points to "17.04.02 12:04:38" (step 1). The time stamp of the online program points to "18.04.02 10:14:22" (step 2). A download of changes is no longer possible, since those two time stamps are not identical.

This scenario shows clearly that a backup copy generated in step 2 would have sufficed to generate identical time stamps in step 4 and thus would have enabled a download of changes.

What situations will prevent online downloads of changes?

It is no longer possible to download changes if the following conditions are present:

- A used block type has been replaced by a new version that contains structural changes, for example, due to the addition of I/Os and/or messages.
- DB and FC number areas have been compressed during compilation using the menu command **Options > Settings > Compile/Download… > Option: "Compress"** DBs and FCs will be assigned a new numbering sequence in this case.
- A modified program was downloaded to another CPU prior to the download of changes, for example, for testing purposes. In this case, the time stamp no longer matches the time stamp of the original CPU.
  Exception: If you use the “Download to test CPU” option in the “S7 Download” dialog box, the download identifier and comparison stamp are retained. You can thus still transfer the program to the original CPU by downloading changes.
- You have restored a program from archive. This is not the original program used for the last download (time stamp comparison).

13.4 System support for avoiding causes of CPU STOP

Introduction

During compilation and download, the system performs checks and evaluations in order to avoid the causes of CPU STOP during online program downloads.

The block reference lists are also used for these checks. These reference lists exist only for blocks compiled with STEP 7 >= V5.0 + SP3 or SCL V5.0 + SP3. This affects blocks of the PCS 7 V5.1 libraries. A full verification is not possible if the program uses blocks without reference lists (blocks of the libraries PCS 7 V4.x and V5.0 and the STEP 7 standard library). In this case, a warning will be displayed.

System support during compilation

The system supports compilation as follows:

- The consistency check includes the entire block call hierarchy and its time stamps. The system can detect that two blocks will call the same block (recursive call), but those called blocks have different versions.
- The maximum local data requirements are calculated and compared with the resources of the configured CPU. A warning is generated if the absolute value (100%) or the warning limit set using the menu command **Options > Customize > Compile/Download…** is exceeded; the code is generated all the same. This allows you to adapt the local data stack without having to recompile the data.
- The program structure is scanned in terms of the nesting depth of its blocks and the result is compared with the maximum possible nesting depth of the configured CPU. Compilation is canceled if the maximum is exceeded. An error message which contains the complete call hierarchy of the relevant OB is written to the log file.
  The maximum nesting depth of an OB is <= 24.
  It is calculated as follows (nesting depth = ND):
  STmax = STmax of OB + STmax of OB 121 + STmax of OB 122
- The OBs are checked to make sure that they call the correct blocks generated by the ES (task FCs). An error message is written to the log file if the incorrect FC is called.
- The number of instance DBs of S7 Communication are counted and compared with the configured maximum number of communication jobs after compilation is completed. The number of communication jobs matches the number of instance DBs of S7 Communication. This check determines whether the set warning limit or absolute limit (100%) is exceeded. A warning message is written to the log file if one of these limits is exceeded.

All error messages prevent subsequent downloads.

System support during downloads

The system supports downloads as follows:

- A check is made to determine whether the connected CPU supports the system functions (SFBs, SFCs) required by the program. The download is canceled if support is lacking.
- The maximum local data requirements calculated during compilation are compared with the resources of the online CPU being used for the download. The system status list is read for corresponding information. If the warning limit is exceeded, the system outputs a warning in the log but does not prevent the download. If the absolute limit is exceeded, the system cancels the download and enters an error message in the log.
- Memory requirements for the blocks to download are calculated and the result is compared with memory resources of the CPU. A check is made to determine whether the set warning limit and/or the absolute limit (100%) are exceeded. A corresponding message is output to a dialog box one of these limits is exceeded. The following options are available in this dialog box:
  - Compressing CPU memory
  - Discarding compression and continuing the download (responsibility of the user)
  - Cancellation of the download

  Memory resources are determined after data compression in CPU RUN mode was completed. The download is carried out if memory requirements are now less than the set warning limit. If the result of compression is not satisfactory, the dialog box appears again with a corresponding message. You can now decide whether to continue or cancel the download.
  If you ignore the warning and continue the download, the download will be canceled on reaching the absolute limit. However, the CPU does not go into STOP. The error message written to the log file.

  Note

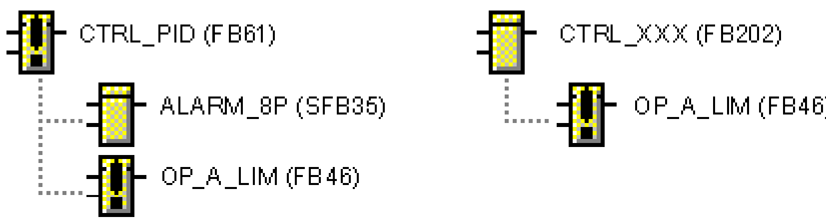
  Note that the calculation of memory requirements cannot ascertain whether existing blocks in the work memory are deleted or overwritten or if so, how many. That is, the actual work memory space required may be less than the data volume of blocks to be reloaded.

  Blocks already downloaded are retained in the CPU if the download is canceled. Only the blocks not downloaded yet are included in the calculation of memory requirements after a new download was initiated. This download procedure can be repeated as required until the entire program was successfully transferred to CPU memory.

- The number of instance DBs for S7 Communication calculated during compilation is compared with the number configured for the online CPU. If the warning limit is exceeded, the system outputs a warning in the log but does not prevent the download. If the absolute limit is exceeded, the system cancels the download and enters an error message in the log.

Additional system support functions

- **Type import**
  The reference lists of the blocks to import are also used during a type import of multiple instance blocks in order to identify the blocks called. This functionality allows you to copy FCs to the target program during import.
  If time stamp evaluation identifies a block which is called by blocks of a different version, the call hierarchy of this block is displayed in a dialog box. Blocks updated during import are identified in this dialog box.
  If a called block is updated implicitly by the import function, another block that calls the same block may no longer be executable.
  Example: The diagram below shows FB 61 which also contains the call of FB 46. Both FBs are updated during import. However, FB 46 is also called by FB 202 which is not updated and therefore unable to access modified parameters.

  From this point onwards, the user must intervene:
  Read the chart reference data by selecting the **Options > Chart Reference Data...** menu command.
  Search for the called block by selecting the **Edit > Find** menu command in the "Block Call Hierarchy" view until all calling blocks have been found. You can also import these blocks after having completed the search.

- **Copy/Move**
  By evaluating the block reference lists, the called blocks will be included if you copy or move multiple instance blocks to another program, similar to the type import.

- **Generate module drivers**
  As PCS 7 process control systems may enter the stop state if errors are detected, it must be possible to react to various causes of errors such as rack/module failures or I/O access errors. To this end, the driver generator inserts the blocks OB_BEGIN and OB_END (the MSG_CSF block was used up to V5.2).
  The error OBs which detect these causes of error and prevent CPU STOP are generated in the CFC and downloaded to the AS.

13.5 Compile and download objects

Overview

The "Compile and Download Objects" function in the SIMATIC Manager allows you to generate consistent modified data of different objects in a single pass, for example, networks, hardware, blocks, charts, OS servers, OS clients, BATCH servers, and BATCH clients. All steps previously distributed across multiple applications are now grouped and processed in the correct sequence.

To initiate the function, select a (multi)project or a station and then select the menu command **PLC > Compile and Download Objects** in the SIMATIC Manager.

Please note that for safety's sake, downloading the entire program using this function is only possible when you have set the respective CPU to the "STOP" operating mode before starting processing.

You can find more information on this function in the STEP 7 Basic Help.

Note

Auto-archiving

With a download using the "Download and compile objects" dialog box, auto-archiving enabled in the "Download" dialog box is not performed.

13.6 Selective download of individual charts

Overview

The "Selective Download" function can be used to download one or more modified CFCs or SFCs to a CPU.

You can use selective downloading as a user to specifically select which configured changes should be transferred to the CPU and enabled. This makes it possible, for example, to configure multiple units on a CPU at one time, because only the changes to a specific unit are compiled, downloaded and tested.

Selecting the charts

- A selection dialog displays all charts that can be selectively downloaded. Select the charts that you want to download in this dialog. Dependent charts are also automatically selected. Dependencies are created by:
  - Inserting the blocks in a sequence
  - Instances of the same SFC type
  - Moving blocks from one chart to another
  - Moving blocks from one runtime group to another
- The charts that are not selected but affected by the changes are transferred automatically.
- System charts with the prefix "@" are not displayed but are automatically transferred as well if they are affected by the changes.

Note on the selection of individual or all changed charts

The selection has the following effects.

- Individual, changed charts are selected and selectively downloaded:
  - Data blocks and user FB/FCs that are no longer used are not deleted in the AS.
  - Likewise, the VXM image is not updated.
  - If necessary, a cross-chart interconnection is replaced by a default value (DB) for selectively downloading when the chart with the interconnection source (output) cannot be downloaded as well.
  - You can find additional information in the paragraph "Notes on process conformity with selective download" below.
- All changed charts are selected in the dialog and downloaded:
  - Data blocks and user FB/FCs no longer used in the AS are deleted.
  - The VXM image is updated.

Note

Booking the PO licenses when selectively downloading

PO licenses are always used for all configured POs when selectively downloading charts, even if only some of them are configured in the charts marked for selective download.

Code change in FB/FC blocks

If a chart is selectively downloaded, and this chart contains an FB/FC block with changed code, then this FB/FC block is also downloaded. The FB/FC block with changed code then affects all other instances in the CPU.

Execution options

The following options are available for executing the "Selective Download" function:

- Component view: In the navigation window, below the desired CPU in the "Charts" folder. This option is used in the description below.
- Plant view: In the navigation window, under the unit in the "Function" folder
  CFCs with different AS assignments cannot be selectively downloaded.
- SIMATIC Manager: Via the menu command "Options > Charts > Selective Download...".
- In the CFC Editor: Via the menu command "CPU > Selective Download...".

Nested charts

This function can only be used on top charts (highest hierarchy level). The subordinate charts of the top chart are always included in the download of individual charts. It is not possible to download individual subcharts.

F systems

Blocks of F-systems are automatically included in a "Selective Download" if they are affected by the changes.

All F-charts are displayed in the dialog if they are affected by the changes and can only be transferred in full. It is not possible to select and download an individual F-chart.

Maintaining the capability to download changes

In order to maintain the capability to download changes, you should create a backup copy after each successful selective download. You can find additional information on this in the section "What you should know about downloading changes (Page 372)".

If PCS 7 Version Trail is also used, the backup copy can be created automatically. To do this, select the option "Archive project after successful download" in the "Compile Program / Download to Target System" dialog.

Chart-based insertion of blocks in runtime groups

Only charts which are integrated in the so-called chart-oriented runtime group management can be selectively downloaded as of CFC V8.2. For this reason, to use the "Selective Download" function at the chart folder, you have to select the "Chart-based insertion" option in the "Properties Chart Folder" dialog. You can find more information on this in the section "Chart-based runtime group management for blocks of CFCs (Page 192)".

When the "Chart-based insertion" option is enabled, this block is automatically inserted into the assigned runtime group when instantiating blocks in a chart. The associated runtime group is created automatically when a new chart is created.

Inserting the new blocks of a chart in the runtime group of this chart reduces the number of dependent charts, which would also need to be included in the selective download.

For PCS 7 projects created as of version PCS 7 V8.1, this option is automatically enabled. For an existing PCS 7 project that was created with a version prior to V8.1, this option only works on newly created charts after the option has been enabled. The existing charts and their configuration are not affected after the option has been enabled.

Notes on the "Chart-based insertion" option

- Effect on runtime groups when renaming charts:
  When the "Chart-based insertion" option is enabled, each chart has a separate runtime group for each OB. The name of the runtime group contains the chart name as a component. If the chart name is changed, then only the corresponding runtime groups in the cyclic interrupts are renamed; all others remain unchanged. However, this violates the rule that the chart name must be a component of the runtime group name, and the clarity is also lost. If the runtime group names have not been previously modified in the runtime editor, all runtime group names must be changed manually when charts are renamed to match the cyclic interrupts.
  Note: If the name of the runtime group is identical to that of the chart, the name of the runtime group changes as well when the name of the chart is changed. A second associated runtime group for the chart is not renamed, however.
- Effect on runtime groups when moving blocks:
  If the "Chart-based insertion" option is enabled for the chart folder, a runtime group is inserted into the run sequence in each affected OB for each chart. Previously, it was only inserted in the cyclic OBs. Blocks that are then instantiated in the chart are automatically incorporated into the runtime group belonging to the chart.
  When blocks are moved to another chart, however, they are not automatically transferred to the corresponding runtime group of the destination chart.
  To maintain the principle of the chart-based insertion and the selective download capability for the changed source and destination chart, you need to manually move the blocks to the runtime group belonging to the new chart.

Notes on process conformity with selective download

With a selective download, the reaction of external interconnections must be taken into account for process compliance:

- Input interconnections (IN/IN_OUT) are active during the download of a CFC following recent configuration. If the source chart with the interconnection source (output) is not included in the download, the required resources are downloaded to the CPU with pre-assigned default values, even if they are not processed in the CPU by the source chart.
- Interconnections to addresses are immediately active according to the configuration (IN/IN_OUT/OUT).
- Interconnections to runtime groups are active if the chart is downloaded with the output interconnections to the runtime group.

- SFC accesses are immediately active, even if the dependent CFCs have changed. If the changed CFC is not yet downloaded, the pre-assigned default values are also active in this case.
- AS-wide interconnections are active across the entire program.

Please read the following safety information "Responsibility for a process-compliant program".

Requirements

- The "Selective Download" function is only available in PCS 7.
- To use the "Selective Download" function, you have to select the "Chart-based insertion" option in the "Properties Chart Folder" dialog at the chart folder so that they are integrated in the so-called chart-oriented runtime group management.
- All charts of this CPU have been compiled and downloaded beforehand. A first-time downloading with the "Selective Download" function is not possible.
- "Selective Download" is not possible if one or more of the following conditions are true:
  - A complete download is required.
  - A complete compilation is required.
  - With the CPU 410-5H PA:
    When the status message "TCiR: Download required" is displayed in the status bar of the CFC Editor. This status message indicates that the instances were updated in the project during interface changes to the block types, but they have not yet been downloaded to the CPU 410-5H PA with the "Type update in RUN" function.
- The icons in the selected charts show the status that these charts can be selectively downloaded. You can find more information on this status above.

CAUTION

Responsibility for a process-compliant program

The "Selective Download" function can be used to download one or more modified CFCs or SFCs to a CPU and to test them. Selective downloading can result in inconsistent data and a non-process-compliant program in the CPU.

The user is responsible for ensuring a process-compliant program.

All modified charts and their dependent charts are displayed in the "Selective Download" dialog to help ensure a process-compliant program. Take this into consideration when selecting charts to be downloaded, because inconsistency in the process flow can arise by downloading a single chart if not all the required/changed charts are included in the selection.

Implementation during ongoing plant operation

The implementation of the recommended actions during ongoing plant operation can lead to serious damage to property or injury to persons if errors occur in the function or in the program.

Ensure that no dangerous situations can occur before you implement the actions.

Please comply with the following:

- If several charts have been changed, for example, with cross-chart interconnections, but not all affected charts are transferred to the CPU with "Selective Download", damage may be caused by inconsistent data.

Procedure

1. Select the "Charts" folder or any CFC in this folder under the desired CPU in the navigation window of the component view.
   Alternatively, you can start the function from the plant view as described above.
2. Select the menu command "Options > Charts > Selective Download...".
   The "Selective Download" dialog opens.
   This dialog displays all charts of this CPU with the icon for their current status.
   System charts with the prefix "@" are not displayed but are automatically transferred as well if they are affected by the changes.
3. Select the desired charts that you want to download to the CPU.
   Select the corresponding check box for this in the "Chart" column. The "Dependencies" column indicates whether other charts need to be downloaded as well.
   The charts that are not selected but affected by the changes are transferred automatically.
   For charts of F-systems, read the information in the "F-systems" paragraph above.
4. Check the settings for generating the module drivers and SCL sources in the lower part of the dialog and change them if necessary.
5. Click "Download".
   The selected charts are downloaded.
   If compilation is needed before downloading the selected charts, this is done automatically.
6. The result of the operation is displayed in the "Logs" dialog.
   Check the displayed messages and close the dialog with the "Close" button.

Result

The selected charts are downloaded to the CPU. The charts that were affected by the changes are transferred automatically.

13.7 Comparing the time stamp of the CPU program

Time stamp

On the basis of the time stamp, you can establish which areas of the user program have been changed and whether it needs to be recompiled or downloaded again. The system will assign time stamps in respect of the following:

- The last download-relevant change
- The last compilation process
- The time when the downloaded program was compiled

Procedure

Select the **CPU > Compare** menu command inthe CFC.

Alternatively, you can also select the SIMATIC Manager menu command:

Select the station or the CPU or the S7 program or the chart folder, followed by the shortcut menu command **CPU > Compare**.

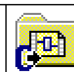
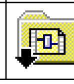
A dialog box showing the current time stamp of the downloaded program appears.

Evaluation

- The "Last download-relevant change" time stamp indicates the time when changes were made to the offline program that also necessitate changes to the online program, for example, interconnection changes or block changes. The time stamp is also changed if block I/Os are programmed in test mode.
  Changes are not classed as download-relevant if they are not taken into account during compilation into machine code. Examples include changes to the position of blocks in the chart or the insertion or deletion of text fields.
- The "Last compilation" time stamp is the time the program was last compiled following a change.
- The "Compilation of downloaded program" time stamp is the time stamp that was generated during the last compilation process and entered at the time of the download. If a connection with the CPU has not been established, it will not be possible to read this time stamp and "---" will be displayed.

The user program that you downloaded to the CPU is only identical with the user program in terms of CFC data management and machine code if the "Last compilation" time stamp and "Compilation of downloaded program" time stamp match exactly and are more recent than the "Last download-relevant change" time stamp.

Tip: You can identify user programs requiring compilation and/or download due to changes by means of the chart folder symbol. In the SIMATIC Manager , the relevant icon is displayed at the closed chart folder.

Compilation required:	
Download required:	

Note on H CPUs

- If the H CPU is being operated in solo mode, for example, due to CPU failure, and a CPU failover has taken place, a dialog box will appear following an online access request There you can select which CPU should be linked. This dialog box does not appear in redundancy mode.
- All changes will be lost if you download program changes to a CPU operating in solo mode and then execute "Switchover with modified configuration" with the menu command **CPU > Operating Mode...**. In this case you need to download the entire program.
  Remedy: Download in redundant operation. However, you must ensure in this case that the operating mode remains unchanged until the download is completed.

13.8 Special considerations during download to a CPU 410-5H PA

Overview

When downloading the program to a CPU 410-5H Process Automation (CPU 410-5H PA), the following special considerations apply:

- The number of POs is licensed by the hardware.
  For the CPU 410-5H PA, the maximum number of POs is licensed by the hardware. It is not possible to download a greater number of POs than is licensed by the hardware into the AS. If this happens, the download operation is canceled.
- The "PCS 7 project" project type is required.
  During download to the CPU 410-5H PA, a check is made as to whether the project has the "PCS 7 project" project type.
  Download is only possible when the project has this project type.
- Download only possible with CFC.
  The CPU 410-5H PA can only be downloaded with CFC V8.0 SP3 (or higher). Downloading from other STEP 7 editors is not possible.
- Monitoring block for license information
  During download to the CPU 410-5H PA, the "@PA-CPU" CFC is automatically downloaded with the "PA_CPU" block. The "PA_CPU" block is required in the CPU 410-5H PA and is used for license monitoring.
- CPU 410-5H PA supports type update in RUN mode.
  This makes it possible to update the instances and download them to the CPU in RUN mode after changing an interface at the block types. This is only possible in STOP mode in other automation systems.
  You can find additional information on this in the sections "How to update block/SFC types in the multiproject (Page 110)" and "Type update with a CPU 410-5H PA (Page 113)".

See also

How to download a user program to the target system (Page 365)

14 Testing user programs

14.1 How to work in test mode

Requirement

Prior to testing and commissioning, the user program created in the CFC Editor must first be compiled free of errors and downloaded to the CPU.

Test settings

You can set the watch cycle under Test Settings. The watch cycle is the cycle time used for updating the I/Os registered for testing.
Default: 2 s.

The watch cycle is CPU-specific; in other words, the setting applies to all charts of the current chart folder. If the current chart is moved to the chart folder of a different CPU, the value set for this CPU applies to the chart.

Setting the watch cycle

1. Select the **Test > Test Settings...** menu command.
   A dialog box opens.
2. Adjust the watch cycle.

In edit or test mode, you can change the watch cycle setting:

- In the active chart window
- In the dynamic display window in which a line of the entries is selected

---

Note

If the cycle time is modified outside the CFC (for example, in SFC), the change is only activated by closing and re-opening the CFC.

Editing modes

The CFC Editor provides two editing modes:

- Edit mode
  In this offline mode you can configure the entire software structure for a CPU.
- Test mode
  Lets you monitor and edit values in online mode.

Note

You can also prepare certain debug functions offline in edit mode, for example, you can add block I/Os to the watch list and edit these.

Test mode

The test mode refers to the CPU to which the currently active chart belongs.

The CFC Editor supports commissioning by providing test functions that allow you to watch and influence the execution of the blocks on the CPU and to change setpoints as required. Testing can be performed in two modes:

- Process mode
- Laboratory mode

In edit mode, you can select the test mode using the menu commands of the "Test" menu. It is not possible to switch between process mode and laboratory mode once you are in test mode.

In process mode, communication for the online dynamic display of blocks is restricted and thus causes only slight extra load on the CP and bus.
All blocks are initiated with the "Watch Off" attribute when you activate test mode.

Laboratory mode allows for convenient and efficient debugging and commissioning. In contrast to process mode, the laboratory mode does not restrict communication for online dynamic update of charts.
All blocks are initialized with the "Watch On" attribute when you activate test mode.

Change log and ES log

Working in test mode, just like downloading to the CPU, is a protected function in S7 that must be logged if the SIMATIC Logon Service is installed.

If access protection and the change log are enabled in the SIMATIC Manager , a dialog box opens when you start and end the test mode and when you change values on the CPU. You enter the reason for your action in this dialog and this is then entered in the change log.

The trackable actions are listed in the ES log if the current chart folder for the ES log was enabled with the menu command **Chart Folder > Object Properties... > "ES Log" tab**.

You can find additional information on this in the following section: Change log and ES log (Page 401)

Activating/deactivating test mode

Activate test mode as follows:

Select the CFC menu command **Test > Test Mode...**or click the following button in the toolbar:

You can now enable the test menu functions; most of the edit mode functions will be disabled.

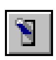
Note

Possible system reactions when activating test mode

- If the user program is modified after the code has been compiled, a corresponding message is output when you want to change to test mode.
- If a so-called protection level is enabled in the CPU, it may be necessary to enter a password when activating test mode depending on the CPU and the configured protection level.

When you **close** the test mode, the test functions will be disabled and the edit mode will be re-enabled.

Exit test mode by selecting the **Test > Test Mode** menu command or by clicking the following button again in the toolbar:



Configuring I/O

You can modify the values of all inputs which are not interconnected if test mode is active. For information on exceptions, refer to the section Essentials for monitoring block I/Os (Page 390). Before the modified value is adopted on the CPU, its validity will be checked.

Note

Note that the parameters modified in test mode are always written to the CFC database.

Notes on H CPUs

- If the CFC is used in fault-tolerant systems (H-CPUs), the CFC in test mode always with the master CPU.
- If the master changes during redundancy mode then the connection to the CPU is terminated. You must exit the test mode and switch the device back on in order to re-establish the connection.
- If the H-CPU is operating in stand-alone mode, for example, after a CPU changeover as a result of CPU failure, any online access (here: Activate Test Mode) opens a selection dialog. There you can select which CPU should be linked. This dialog box does not appear in redundancy mode.

14.2 Monitoring and assigning parameters to block/chart I/Os

14.2.1 Essentials for monitoring block I/Os

Monitoring block I/Os

If "Watch" is enabled for I/Os that have been registered for debugging, the CPU will provide these I/Os with current values in debug mode.

When you start test mode, the "Watch On" function is also enabled for CFC charts in "laboratory mode" with the following icon:

This means, in test mode you can monitor dynamic data of block/chart I/Os registered for testing, i.e., those values will be fetched cyclically from the CPU and displayed. and you can modify the options for this dynamic data display and the parameters of the I/Os with OCM capability.

In "process mode", the "Watch Off" function is enabled with the following icon:

That means, in a first step you must select the blocks/charts for dynamic data in the chart you want to watch. Then, select "Watch On".

If, on the other hand, there is an overload, you can remove specific blocks/charts from the watch list. To do this, select the block/chart followed by the menu command "Watch Off".

Note

If monitoring is disabled and you select an I/O in order to register it for testing, monitoring will be enabled for **this** and for **all previously registered** I/Os of this block.

Monitoring of an I/O is enabled with the following icon:

When you enable test mode for an active dynamic display window, all I/Os listed will be connected to the CPU. Enable the "watch" function with a check mark in the "Watch" column for the specific I/Os.

Note

Please observe the following information:

- You cannot watch block I/Os if they are not stored in DBs. This also applies, for example, to open inputs of FCs and BOPs and to outputs of the data type STRING, DATE_AND_TIME, ANY.
- In test mode and if EN=0 is set, the function blocks (FBs) do not indicate the value of the connected source at interconnected inputs. Those interconnected inputs will only display the value with which the block was last executed. Note that the value of interconnected inputs can change when EN=0 is set to EN=1. This statement also applies to forced values. FCs and BOPs, on the other hand, always show the value of the interconnection source.
- If I/Os are registered for testing but are not operable, for example, if they are interconnected, the values are shown with a yellow-gray background.
- You can display the value of a block I/O as a tooltip even if the I/O has not been registered for testing. See also: Displaying tooltips using the cursor (Page 444)
- In test mode, only the status of the first element of a structure is shown in color. The status of the other elements is not visible.

Additional information

You can find additional information about this in the following sections:

- How to add and remove I/Os in the watch list (Page 391)
- How to enable/disable monitoring of dynamic I/O data (Page 392)
- Configuring I/O (Page 393)
- The "Dynamic display" window (Page 397)
- The trend display window (Page 399)

14.2.2 How to add and remove I/Os in the watch list

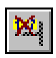
Add to watch list

In edit or test mode (process or laboratory mode), you can add individual block or chart I/Os for debugging. Select the I/O followed by the menu command **Debug > Add I/O** or click the following button on the toolbar:

When you change to test mode, monitoring is also enabled, i.e., the I/O and its current value are displayed on a yellow background. With I/Os that are not OCM-capable, the value is shown on a yellow-gray background. If the watch function was previously disabled for this block/chart, it will be enabled for all other I/Os that were previously added to the watch list.

Remove from watch list

You can remove I/Os from the watch list by selecting the I/O followed by the menu command **Test > Remove from Watch List** or by clicking the following button on the toolbar:



This is only possible in test mode and if "watched" is selected.

As an alternative:

In edit mode, you can also add and remove I/Os for a specific block/chart. Call the "Object Properties" dialog for a selected block and select or deselect the individual I/Os in the "Watched" column of the "I/Os" tab.

Note

Do not watch too many I/Os at the same time, for this would induce a high communication load on the bus and on the CPU. Time monitoring may respond if this load exceeds a certain capacity, for example, >500 I/Os in a 1-second watch cycle. The I/O values will be temporarily marked as "corrupted".

14.2.3 How to enable/disable monitoring of dynamic I/O data

Enabling and disabling

You can enable monitoring of block/chart I/Os in the watch list as follows:

- Automatically, by enabling the debug mode in "laboratory mode" for all blocks
- with the menu command **Debug > Watch On** or by clicking the following button in the toolbar:

  In laboratory mode, this function applies to all blocks and in process mode only to blocks previously selected in the chart.
- You can stop the monitoring function using the menu command **Debug > Watch Off** or by clicking the following button in the toolbar.

  The values of the I/Os are no longer updated In laboratory mode, this applies to all blocks and in process mode only to blocks selected in the chart.

All I/O values activated in the watch list will be updated in accordance with the watch cycle set using the menu command **Debug > Test Settings...**.

The values are displayed beside the I/Os according to their data type. They are displayed on screen on a colored background.

Blue on white	Display of the values in the edit mode (offline)
Black on yellow, asterisks	Display of the values during the transition to dynamic display
Black on yellow, value	Display of the values fetched from the CPU during test mode
Black on gray-yellow, value	Display of values at I/Os without OCM capability read from the CPU in test mode
#### on a red background	Display of the values when the dynamically updated values requested from the CPU cannot be supplied (disturbance, overload)

14.2.4 Configuring I/O

You can modify the values of all inputs which are not interconnected if test mode is active. For information on exceptions, refer to the section Essentials for monitoring block I/Os (Page 390). Before the modified value is adopted on the CPU, its validity will be checked.

Note

Note that the parameters modified in test mode are always written to the CFC database.

14.3 Forcing block I/Os

Brief description

It may be helpful during commissioning to simulate a variety of values for an interconnection, in other words to permanently overwrite the value of an interconnection with a force value. While the configuration of an input in test mode is basically possible, it will not lead to the desired results because the value will be immediately overwritten by the output of the interconnection. For simulation, the interconnections between the blocks must be temporarily removed and specified values must be assigned to the corresponding inputs (IN or IN_OUT) of these interconnections. Replacing an interconnection with this type of configuration is hereafter referred to as "forcing". Forcing involves replacing the value at the block input normally supplied by the interconnection with the "forced value". Such forcing can be activated and deactivated at the input of the block instance at any time.

For reasons of performance, not all block inputs are designed for forcing at the outset. This prevents too much SCL code being generated by the code generator. The inputs that can be forced are set in the configuration of the CFC or in the process object view. If the attributes "Add forcing" and "Forcing active" are changed at the input after the program is compiled, the program needs to be compiled and loaded again.

A maximum of 8192 standard and 8192 F I/Os can be forced.

Note

If the maximum number of force parameters is registered and downloaded, a download of changes must be carried out after one or more parameters have been disabled by Force in order to re-enable the parameters in the CPU as well. New or different force parameters can be enabled for Force only after the download of changes.

Settings for forcing

Forcing is controlled by 4 attributes:

- "Support forcing"
- "Add forcing"
- "Forcing active"
- "Force value"

The use of these attributes is enabled with the corresponding check boxes in the SIMATIC Manager and CFC.

In the SIMATIC Manager in the object properties of the chart folder in the "Advanced" tab:

- "Support forcing" check box This enables the force function and the corresponding options in the CFC and the process object view.

In the CFC in the object properties of the block input:

- "Add forcing" check box This enables or disables "Forcing" at this input. Each change requires the program to be compiled and loaded again. This option cannot be changed in test mode.
- "Forcing active" check box When this check box is activated, the value of the interconnection is permanently replaced by the force value. The value of interconnection becomes active again when forcing is disabled. A change in test mode does not require recompiling.
- "Force value" text field Enter a value here to be applied to the block input if the options "Add forcing" and "Forcing active" are enabled. A change in test mode does not require recompiling. At an INOUT, the force value is also written to the output of the interconnected block.

Alternative procedure

If the "Support forcing" option is activated for the chart folder, you can proceed as follows:

1. You can make settings for multiple block inputs in the CFC. The corresponding columns for the force function are available in the "I/Os" tab of the object properties of the block.
2. In the process object view, you can make the setting for the desired inputs for all blocks in the project. The corresponding columns for the force function are available in the "Parameters" and "Signals" tabs.

Display

The interconnection of the forced input is identified in the **CFC** by means of a colored rectangle at the block input.

- A green rectangle means: "Add forcing" is activated
- A red rectangle means: "Add forcing" **and** "Forcing active" are activated.

Note

Colored rectangles are only visible for interconnection, as forcing is only possible for interconnected parameters.

In test mode, the force value is distinguished from the other dynamic values by a different background color. The default setting is "light blue" and can be changed in the "Color Settings". Only the first element of a structure is shown in color. Other elements are not visible.

The background color of the force value is identical to the **dynamic display** of the chart.

Note

All force settings are lost after a cold restart of the CPU is performed while forcing is activated. The settings are retained in the offline program. To restore consistency between the offline and online programs, disable "Support forcing" at the chart folder, compile and download the data, re-enable "Support forcing" at the chart folder and once again recompile and download the data.

Although connections with textual interconnections can be registered for forcing, this does not have any effect in test mode.

Message to WinCC with forcing enabled (only when using an S7-400 CPU)

In the case of forcing, a new system chart @FRC_CFC is automatically installed with a runtime group of the same name in OB1 during compilation. The message block FRC_CFC is added to this chart, as well as being added to the OB100. This block triggers an active message to WinCC if "Forcing active" is set at a parameter. The block triggers a corresponding outgoing message after "Forcing active" was disabled again. The "Active" control option of the @FRC_CFC runtime group defines that the block is only executed after the "Forcing active" function has changed.

If forcing is disabled, the block, the system chart, and the runtime group are removed again from the program the next time you compile and download.

Data types

The following data types can be forced:

BOOL, BYTE, INT, DINT, REAL, STRUCT, WORD, DWORD, DATE_AND_TIME

With the STRUCT data type, only the first level of the structure can be forced. Chart inputs/outputs cannot be forced.

Note

If an EN input or the input of an FC or BOP is registered for forcing, the parameters can be set for this input in test mode making it capable of operator control and monitoring.

14.4 The "Dynamic display" window

Dynamic display window

In debug mode, the values of block and chart I/Os can be displayed dynamically in a separate window. This is possible for the basic data types (BO, W, R, etc.) and for elements of structures.

The dynamic display window can be opened and arranged in the CFC Editor window along with any chart windows and/or the trend display. The window size is adjustable. The content (static content only, not the values) of the window is saved when you close it or when the CFC Editor is exited and restored when you open it the next time.

You can create and manage any number of dynamic displays for each CPU. Each display has its own name that you enter when you create it and that you can modify at any time. In the dynamic display window, only one dynamic display can be active at any one time. You make your selection in the combo box in the header. This bar also contains the buttons for creating, deleting and renaming dynamic displays.

In this window, the values of I/Os from different charts of a CPU can be displayed, monitored and changed. The relevant chart does not need to be open. The connection to the CPU must be established.

Use the menu command **View > Dynamic Display** to open and close the dynamic display window.

Inserting I/Os

You can add I/Os to the dynamic display in edit or test mode. This function is handled in the same way as the one for adding or removing block I/Os in the watch list.

How to add I/Os of an open chart to the dynamic display window:

1. Open the required dynamic display in the dynamic display window (relevant only if more than one dynamic display was created).
2. Select the desired I/O, and then select the menu command **Debug > Inputs/Outputs > Insert in Dynamic Display**.
3. Drag-and-drop this I/O to the dynamic display window.

For block I/Os with a structure: Before the I/O is entered in the dynamic display, a dialog box appears in which you can select the structure element for the value to be displayed.

If you want to add all I/Os of a block or nested chart to the dynamic display, you can select the block or chart and drag-and-drop it to the dynamic display window.

I/Os of the data type STRUCT and hidden I/Os will be excluded.

Deleting I/Os

To delete I/Os, select the relevant lines in the dynamic display and delete them using the shortcut menu command **Delete** or by selecting the menu command **Edit > Delete**.

Note

If you delete blocks whose I/Os are contained in a dynamic display or move them out of the S7 program, the relevant entries will also be removed from the dynamic display.

Monitoring values

You can watch block/chart I/O values without opening the relevant charts.

The value is displayed dynamically on a yellow background according to the set watch cycle if:

- Test mode is enabled.
- The I/O or element of a structure is registered for monitoring via selection in the "Watch" column.
- Your station is connected to the CPU.

Corrupted values are indicated with "####", an I/O value that cannot be monitored is displayed as "- - - -".

A forced value is not shown with another background color as yellow (as specified in the "Color Settings"). The default setting is "light blue".

Printing

You can also print the contents of the dynamic display. Use the menu command **Chart > Print ...** or click the printer symbol in the function bar to print the values that were current when you started printing.

Open Chart

If you select a row in the dynamic display, you can jump to the chart that contains the block whose I/O is monitored. Call menu command **Edit > Go To > Chart** to open the relevant chart and to highlight the I/O.

14.5 The trend display window

Trend display

With the trend display in CFC, you can follow the changes in one or more signals of a CPU continuously over time. The trend display can be used by all target systems that support normal online operation. A maximum of 12 values can be recorded at one time.

Note

If you delete blocks whose I/Os are contained in a trend display or move them out of the S7 program, the relevant entries will be removed from the trend display.

You can create and manage any number of trend displays for each CPU. Each display has its own name that you enter when you create it and that you can modify at any time. In the trend display window, only one trend display can be active at any one time. You select trend display in the drop-down list in the "Trend Display" box in the header. This bar also contains the buttons for creating, deleting, renaming, and exporting trend displays.

Use the menu command **View > Trend Display** to open and close the trend display window.

- When you open the trend display for the first time, an empty window is displayed with the name "Untitled". You can rename this trend display.
- If there is more than one trend display, the display at the first position in the drop-down list is displayed when you open the trend display following a restart of the CFC.

Operator inputs and settings

You must assign each I/O for which you wish to record values to one of the 8 trend display channels. You can only assign I/Os with numeric data types (BYTE, INT, DINT, WORD, DWORD, and REAL) and the Boolean data type (BOOL).

For each selected channel with numeric data type, you can set the high limit and low limit for the display parameter (y axis). Assignments that are no longer required can be deleted for the specific channel.

You can jump from an occupied channel to the chart containing the I/O used. The chart is opened and the relevant I/O is selected or, if the I/O is interconnected, it flashes.

Enter the number of measuring points to be displayed (x axis) for each channel in the input field. The number can be between 10 and 500 and is effective after you click "Apply". The number specified here cannot exceed the buffer length (for additional information, see "Recording").

During the recording, the recording time (start and end) is not displayed but is replaced by the text "Recording...".

Click "Change..." to open the "Recording Parameters" dialog box and specify the following:

- The number of values to be recorded (buffer length)
- The acquisition cycle (in seconds)
- Recording mode ("Continuous", "Continuous with abort conditions", "One-time")
- Abort conditions

The "Start" button is active only in test mode. After starting, the start button changes into a “Hold” button and is relabeled accordingly. You can then use it to stop recording at any time.

Recording window

After starting, the last recorded trend is deleted and the values acquired now are displayed. If the display area is smaller than the total number (buffer size), the window scrolls automatically when the right edge of the window is reached so that the latest value is always displayed.

If you record continuously, the display is shifted left when the maximum buffer size is reached.

The display of the entire time axis below the recording window depends on the number of values (buffer size) entered for "Recording" and the acquisition cycle.

The visible time axis and, therefore, the scale, are decided by the sampling values entered for "Display" and the acquisition cycle.

If the buffer length is greater than the number of sampling values for "Display", a slider is displayed below the time axis. You can use it to move the visible section in the window.

Printing

It is not possible to print the trend display directly. You can, however, export the current trend display in a format that can be read in Excel, for example, and then edit the display in this program and print it as a table. To export the display, click "Export...". You should check the settings for export using the menu command **Options > Customize > Export Trend Data**.

Change log and ES log 15

Introduction

Various logs for tracking changes are available in the engineering system.

- Change log
- ES log
- Log in the "Download" tab

The change log

The change log is opened when an action is performed that needs to be logged if the following conditions are met:

- The SIMATIC Logon Service is installed.
- Access protection and the change log are activated in the SIMATIC Manager.

The following protected actions are logged:

- Download to PLC (entire program)
- Download to PLC (changes only)
- Activating and deactivating test mode
- Changing values in test mode

The change log records the user, time, CPU, changes made, and the reason for the changes.

If you have not enabled the change log in the SIMATIC Manager, you cannot edit the reasons for the change.

Open the change log in the SIMATIC Manager (with a SIMATIC station selected) with the menu command **Options > Change Log > Display**.

In the list of log entries, you can find the change log for CFC/SFC with an entry in the format "YYYYMMDD_hhmmss" in the "Work object" column.

The ES log

In addition to the information of the change log, the actions with time stamp relevant for the downloaded can be logged in the ES log in the CFC/SFC.

Requirement:

- The ES log must be activated for the respective chart folder.
  A description of the activation and display is available below.

Activating the ES log

To enable the ES log for the current chart folder, proceed as follows:

1. Select the menu command **Object Properties**.
2. In the "Properties Chart Folder" dialog box, select the "ES log active" check box in the "Advanced" tab.
   The ES log is activated.

If protected functions do not need to be logged, for example, within the context of initial configuration, you can deactivate the ES log. To do this, clear the "ES log active" check box.

Note

Please observe the following information:

- If you are not using an S7 target system, check in the target system online help to find out whether the ES log is supported by the target system.
- Note that an activated ES log can only be deactivated on the computer on which SIMATIC Logon Service is installed. Reason: Deactivation, as well as activation must both be logged.
- If you copy the program or chart folder with an active ES log to a computer on which the SIMATIC logon service is not installed, you will receive an error message while downloading or switching to test mode, and the action will not be executed.
  Deactivating the ES log in this case is not possible because the "ES Log" tab is not offered in the object properties dialog box for the chart folder. Activation and deactivation can only be set in this dialog box.
- If access protection and the ES log are activated, the SIMATIC Manager’s "Compile and Download Objects" function is interrupted prior to the download to each individual CPU with the opening the ES log.

Logging in the ES

ES log

Open the ES log in CFC with the menu command **Options > Logs... > Tab: ES Log**.

Each action is recorded in the log in a main line, followed by any additional relevant details of the action (for example, a download log). The actions are logged sequentially in chronological order.

When the action "Download entire program" is performed, the ES log is cleared but simultaneously saved as a file with a date ID. The archiving action and the file name used (including the path) are recorded in the log.

During download, the comparison stamp of the target system is also written to the log:

- Last download-relevant change
- Last compilation
- Compilation of the loaded program

When the "Test mode on" action is performed, the subsequent actions resulting in a change in values in the CPU are logged:

- In CFC:
  - Configuration of the connections
  - Activation/deactivation of runtime groups
- In SFC (PCS 7):
  - Configuration of constants in steps
  - Configuration of constants in transitions
  - Configuration of constants in sequencer properties

The logging includes the value and how it changed (address, old value, new value).

Log in the "Download" tab

- Download log in text format
  When the change log is disabled, information about the download is logged and stored in text files which are named in "YYYYMMDD_hhmmss.log" format.
  In the "Download" tab of the "Logs" dialog box, the "Archive" button allows you to open a dialog for selecting and displaying these log files.
  When the change log is deactivated, the "Archive" button is deactivated.
  Reference in SIMATIC logon:
  A reference to the associated download log file is made in the SIMATIC Logon log. In the list of log entries, you can find the reference to the download log for CFC/SFC with an entry in the format "YYYYMMDD_hhmmss" in the "Work object" column. You can find and display the named log file using the "Archive" button in the "Download" tab.
- Download log in XML format:
  The download log can also be saved in XML format. The download log contains information in XML format on all objects to be downloaded.
  To generate the download log in XML format, the "Generate image of downloaded program for comparison" option must be activated in the CFC editor in the "Settings for Compilation/Download" dialog box. The "Settings for Compilation/Download" dialog box is used to open the menu command **Options > Customize > Compile/Download**.
  This "Generate image of downloaded program for comparison" option is automatically activated when the change log is activated.
  Depending on the change log:
  - Change log is not activated:
    The download log in XML format is temporarily created by default under the name "LOADOBJECTS.xml" and overwritten during each compilation run.
    Before downloading to the AS, this log can be displayed using the "Display changes" button in the "Download target system" dialog box.
  - Activating the change log:
    After each download, the download log is stored in XML format and name in "YYYYMMDD_hhmmss.xml" format. You can find the named log file using the "Archive" button in the "Download" tab.
    A corresponding line with a reference to this XML file is entered in the log in the "Download" tab.
  - The XML files with the information about the downloaded changes can be compared in the VXM if required.

Reading back charts 16

16.1 How to save parameters (read back)

You can save (read back) the parameters of all CFCs contained in the chart folder of the active chart whose AS data were, for example, changed via OS operations. This applies to, for example, limit values or control parameters.

Note

You will automatically be presented with a read-back option before a complete download of the program is performed. This is aimed at preventing values previously modified by you in debug mode or on the OS from being lost unintentionally if all blocks are deleted. You are free to accept or reject the read-back option, which is made available in a dialog box.

Read-back is only possible if no interface changes have been made to block types, i.e., provided that a download of changes remains possible.

Procedure

1. Select the menu command **Chart > Read-back...** in CFC Editor, or select the menu command **Options > Charts > Read Back Program...** in SIMATIC Manager.
2. Select the source in the "Read Back" dialog box:
   - Select "Program on the CPU"
     to read back the program with the current parameters from the CPU (online block folder) into the chart folder.
   - Select "Program offline"
     if you do not have direct access to the CPU and have copied the online block folder into the offline block folder of an S7 program in the plant. A browser will then open before the read back allowing you to select the respective S7 program with the plant data. The values are then read from this source and written to the ES database.

You can select the following in the dialog box in both cases:

- All parameters of the block inputs
- OCM-capable parameters (system attribute S7_m_c: = 'true')
- Designated parameters (system attribute S7_read_back := 'true')

The default setting for the scope is "OCM-capable parameters".

General rules

The following applies regardless of the selected scope:

- Only in/out and input parameters can be read back. Output parameters cannot be read back.
- The following data types are not included in the readback:
  ANY, ARRAY, POINTER, UDT
- Inputs, which write to the SFC, are not read back.
- If a parameter is never to be read back, you must set the attribute `S7_read_back = never`.
  If a parameter is always to be read back, you must set the attribute `S7_read_back = always`.
- Entire blocks can be excluded from the readback, for example, BATCH blocks. In this case, the block type contains the attribute S7_read_back := 'false'. The attribute can be changed at the block instances with the menu command **Block Properties > "Read-back enabled" Option** .
- Unconfigured inputs (S7_param = "false") are not read back if they have a simple data type. S7_param = "false" does not prevent readback for complex data types.
- **Note for H-CPU**
  When the H-CPU is in solo mode, due the failure of a CPU for example, and the CPU has switched over, a dialog box opens for online access (in this case: Readback). There you can select which CPU should be linked. This dialog box does not appear in redundancy mode.
- **Note about downloading HW Config data**
  After the download of HW Config data, note that the data is taken from the loading memory. Changes performed by operator control and monitoring or in test mode are only available in the main memory. The changes in the main memory are lost when the data is taken from the loading memory because it is overwritten by the initial values of the loading memory. If you have performed a readback before downloading the HW Config data to obtain the operator values, you can transfer it to the CPU with a complete download in the STOP state.

Rules for the scope

The following applies for specifying the scope as "OCM-capable parameters" und "Designated parameters":

- Values from FC inputs are ignored.

The following applies when specifying the scope as "All parameters":

- FC inputs that are interconnected with chart I/Os are read back. This also applies to chart I/Os that are multiply interconnected to block inputs (FB and FC).

After the readback

- A complete compilation is not required after the readback. The program can also continue to be downloaded when the CPU is in the RUN state (download of changes).
- A log is generated and displayed when the readback is completed. You can later open the log with the menu command **Options > Log > "Read Back" tab**. The following is displayed in the log, for example:
  - Which I/Os were read back and changed
  - Where problems occurred

17 Signal processing

17.1 How to generate module drivers

Generating module drivers automatically

PCS 7 provides a signal processing function that automatically generates required module drivers, interconnects them and sets their parameters accordingly after you have configured the hardware using HW Config and the technological functions in CFC. These module drivers are required for diagnosing and reporting errors during signal processing.

The function is called when you compile the program if the "Generate module drivers" check box is enabled (default). If module drivers have already been generated for the project, the system checks whether they need to be updated during processing. An update is necessary if the hardware configuration has changed in the meantime.

Generating module drivers manually

You can also call the function from the SIMATIC Manager . Proceed as follows:

1. Select the chart folder.
2. Select the menu command **Options > Charts > Generate Module Drivers...**.

In the following we shall refer to the "Generate Module Drivers" function as "driver generator".

The driver and message concept described here is only applicable to CPUs of the S7-400 family.

Note

If the address spaces for digital I/O modules were packed in HW Config ("Pack addresses" function), the driver generator can no longer supply the associated blocks with unique addresses. To ensure that each module has a defined slot assignment, the addresses must not be packed.

Driver and message concept

An extended driver concept was introduced with CFC version V5.2 + SP1. The driver generator can be executed either according to the extended or previous concept. The concept actually used for existing projects depends on the signal-processing blocks (CH blocks) used. Imported block types are searched and the version of the CH blocks is checked. The previous concept will be applied to CH blocks of a version < 2.0.

You can find additional information on this in the following section: Signal processing with driver concept up to V5.2 (Page 446).

The extended driver concept

The extended driver concept supports additional modules for ET 200M, ET 200S, ET 200X, PA devices, DP/PA couplers, DP/PA links, Y links, DPV0/DPV1 slaves, diagnostic repeaters, HART field devices (diagnostics), and redundant I/Os released for PCS 7. This enables the precise identification of channel errors in messages. It is also possible to apply configured data from SIMATIC PDM.

The new nested interconnection model (Page 414), module drivers for driver blocks in combination with the OB_BEGIN and OB_END blocks for CPU and connection diagnostics (instead of MSG_CSF with V5.2) provides the means for runtime optimization.

This also leaves an opening for future expansions. When implementing user-specific and third-party module types, the meta-knowledge for the driver generator can be expanded by additional XML files (object and action lists). How to create those files is described in the *Programming Guide for Creating Driver Blocks for PCS 7* manual.

Note

Please observe the following information:

- The driver block library must be installed on your PC via Setup. Only this type of installation ensures that sufficient meta-knowledge is available for the driver generator. You are not permitted to copy the library from one computer to another.
- You can also use driver blocks from other libraries (for example, user blocks from a custom library). In the "Generate Module Drivers" dialog box you can specify an additional library. The driver generator then looks in this additional library first for every block to be imported. It will only look in the control file (XML file) when it cannot find the block in the specified library.
- If the S7 program contains a block for processing signals (CH_xx, CH_U_xx, PA_xx), but none from one of the PCS 7 libraries, you must specify the version of the driver library from where the driver blocks should be imported in the "Generate Module Drivers" dialog box.
- The signal processing blocks (CH_xx, CH_U_xx, PA_xx) must be interconnected with a corresponding signal name from the symbol table. This is the only way to ensure that the driver generator considers this block during generation of the system charts and parameter assignment of the blocks for signal processing. Otherwise, the functionality of other signal processing blocks with the same parameter assignment may be affected.

Additional information

You can find additional information on this in the following sections:

- Brief description of blocks used (Page 412)
- How the "Generate Module Drivers" function works (Page 415)
- Session model for "Generate Module Drivers" (Page 417)
- Devices supported by the driver generator (Page 411)

17.2 Devices supported by the driver generator

Table 17-1

Catalog folder	Supported devices
SIMATIC 400	All racks and SM modules in the HW Config catalog (profile PCS 7_V52 or PCS 7_V60 or PCS 7_V61 or PCS 7_V70 or PCS 7_V71)
SIMATIC PC station	All racks and SM modules in the HW Config catalog (profile PCS 7_V52 or PCS 7_V60 or PCS 7_V61 or PCS 7_V70 or PCS 7_V71)
PROFIBUS DP	ET 200iS, ET 200M and ET 200X slaves (as of V6.0 including ET 200S) • All racks and SM modules listed in the HW Config catalog (profile PCS 7_V52 or PCS 7_V60 or PCS 7_V61 or PCS 7_V70 or PCS 7_V71) • SM modules inserted on the slaves • CP and FM modules, with certain restrictions, see the list "PCS 7 - Released Modules". • With ET 200S, also PM and motor starter, with certain restrictions, as of V6.1 also downstream from Y link, see the list "PCS 7 - Released Modules". Non-redundant standard slaves: For DPV0 slaves, a diagnostic block OB_DI-AG1 is placed. As of V6.0 also for DPV1 slaves. As of V6.0, diagnostic blocks are placed for the diagnostic repeaters.
PROFIBUS PA	For all PA slaves acc. to DPV0 (slave family 12 and PA profile 3.0 required) configured downstream from a PA link, appropriate PA diagnostic blocks are placed (access via the signal-processing blocks PA_xx or CH_U_xx). As of V6.0, also PA slaves acc. to DPV0 and DPV1 configured directly on a DP master system over a DP/PA coupler. PA slaves acc. to DPV1 downstream from a PA link (DPV1) are also not processed. PA devices with a maximum of 16 allocated slots are supported.

Note

In addition, device manufacturers may deliver their own diagnostic blocks with corresponding XML files, which are then processed by the driver generator.

17.3 Brief description of blocks used

Block Types

The following block types are used, which separate the hardware and software configuration:

- The user inserts channel-specific blocks (CH/PA blocks) into the CFC chart and interconnects those entries to corresponding signal names from the symbol table. These blocks and system functions are always included in the same process, since the blocks are a part of system functions.
  The driver block library provides the following types of channel blocks for signal processing:
  - **Standard channel blocks**:
    CH_AI, CH_AO, CH_DI, CH_DO.
    These blocks are only used to process signals of the S7-300/400 SM modules. Use these standard blocks if you want to optimize memory utilization and runtime and do not need to process any PA devices.
  - **Universal channel blocks**:
    CH_U_AI, CH_U_AO, CH_U_DI, CH_U_DO.
    These blocks are used for processing the signals of S7-300 / 400 SM modules or PA field devices. The advantage of these blocks is that you can create CFC charts irrespective of the hardware I/O to be used later. A disadvantage is an increase of the load on memory and cycle times.
  - **PA channel blocks**:
    PA_AI, PA_AO, PA_DI, PA_DO, PA_TOT,
    These blocks are designed especially for use with PA field devices. They are used primarily where the special features of these devices are required. In contrast to CH blocks, the PA channel blocks process not only the actual signal but also all variables, according to the desired device configuration selected in the hardware configuration.
  - **Special channel blocks**
    CH_CNT, CH_CNT1, CH_MS.
    These blocks are required for special applications such as controlling and reading the counter or frequency values of FM 350-1/-2 modules and 8-DI NAMUR modules of the ET 200iSP, as well as for signal processing of ET 200S motor starter modules.
  - **Standard channel blocks in the Advanced Process Library:**
    Pcs7AnIn, Pcs7AnOu, Pcs7DiIn, Pcs7DiOu, Pcs7DiIT
    These blocks are used only for processing the signals of S7-300/400 SM modules. Use these standard blocks if you want to optimize memory and runtime utilization and do not need to process any PA devices.
  - **FF/PA channel blocks in the Advanced Process Library:**
    FbAnIn, FbAnOu, FbDiIn, FbDiOu
    These blocks are specifically intended for use with PA field devices and the PROFIBUS 3.0 class A and B with FF field devices. You should use these blocks if you want to make use of the special features of these devices. In contrast to standard channel blocks, PA channel blocks not only process the signal itself but also all variables, according to the desired device configuration selected in the hardware configuration.

For additional information about the functions and functionality of CH and PA blocks, refer to the context-sensitive help (<F1>) for each block.

- In the extended driver design, MSG_CSF was replaced by the OB_BEGIN block for standard CPUs and H CPUs. This function monitors the status of the (redundant) CPU and the DP master systems. It fetches OB startup information (for more on this, see this figure: Interconnection model, module drivers (Page 414)), determines the geographical address based on the logical address and passes this information to the output (CPU_DIAG structure). All driver blocks are interconnected, using this structure. OB_END is installed at the end of the sequencer of all OBs and resets startup information.
  OB_BEGIN also performs connection diagnostics, for example. Process control enclosures, which were monitored with MSG_CSF, are no longer monitored. This function must be implemented in a separate block as required.
- Using OB 70 and OB 86, the SUBNET blocks analyze the status of the DP master systems and return this information (linked to the CPU monitoring) as higher-ranking errors to the RACK blocks.
- The RACK block monitors the status of a rack, DP/PA or Y link DP V1 and reports the error events.
  In order to prevent all MOD blocks from reporting a module failure if a rack fails, a RACK block in each rack takes over the reporting function. The MOD blocks recognize in OB 86 that the event has been processed and do not report.
- Using diagnostic blocks:
  - MOD blocks are used to monitor SM modules, motor starter modules and HART/PA field devices. They provide the channel-specific value status for the signal-processing blocks (1 = good, 0 = bad). ALARM_8P is used to report these events. Errors originating from a higher level, for example, DP master system diagnostics, are entered in the RACK block.
  - DP/PA slaves are monitored with the PADP_L0x (V0) or PADP_L10 (V1) blocks, PA slaves with the DPAY_V0 or DPAY_V1 blocks.
- The PO_UPDAT block ensures that the startup values that were written to the process image by the CH_AO and CH_DO blocks are sent to the output modules when the CPU is restarted (OB 100). This means that these values are effective immediately when the CPU changes to RUN.

For additional information on the functions, functionality and message capability of the blocks, refer to the context-sensitive help (<F1>) for the block.

Additional information

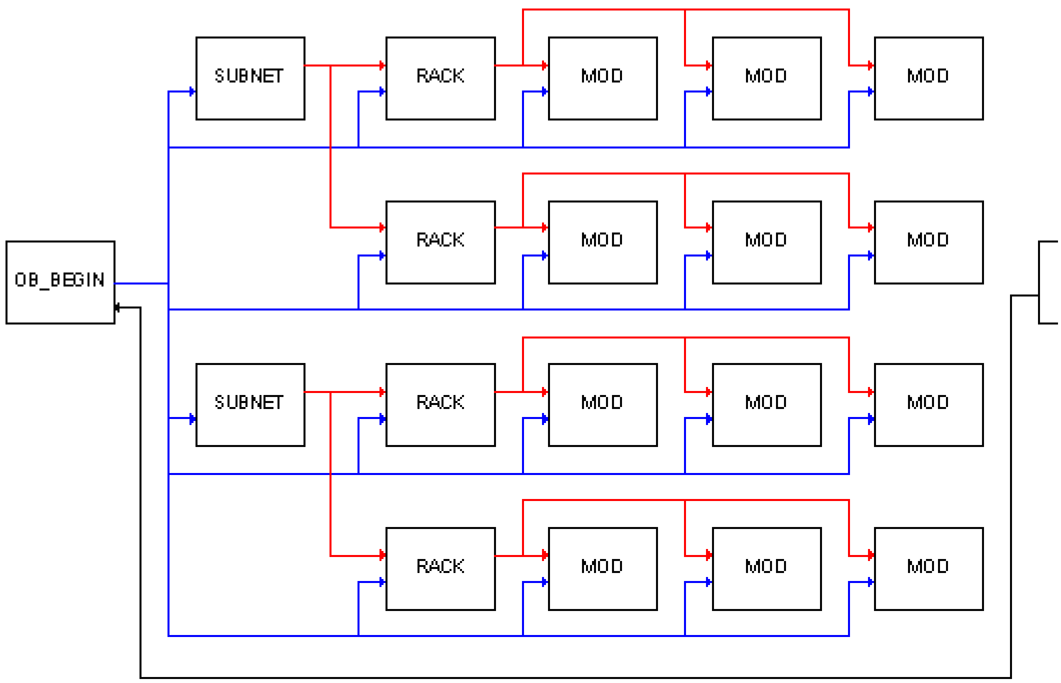
You can find additional information on this in the following sections:

How the "Generate Module Drivers" function works (Page 415)

Session model for "Generate Module Drivers" (Page 417)

Devices supported by the driver generator (Page 411)

17.4 Interconnection model, module drivers



17.5 How the "Generate Module Drivers" function works

How it works

This function automatically generates system charts and inserts the diagnostic blocks (MOD or PADP/DPAY blocks), the RACK and SUBNET blocks, OB_BEGIN and OB_END, and the block for outputting the process image PO_UPDATE. System charts are assigned the name "@.....".

Blocks inserted by the driver generator are assigned the name of the module assigned. Process control messages of the WinCC message system are assigned the name of their origin. In this case, this is the block name and thus the name of the monitored module.

Tip: You can assign a specific text to process control messages by setting a corresponding module name in HW Config. This text then becomes part of the process control message.

The blocks are imported from the library when you initially generate module drivers and the CFC data management does not yet contain any driver blocks (including OB_BEGIN and OB_END). The block search starts at the path defined in the "Settings – Generate Block Drivers" dialog box. If a block is not found in this path, the searches extendss to the default paths defined during setup of the PCS 7 libraries.

Note

Please observe the following information:

- Objects identified by the "@" character should not be modified by the user, but should only be manipulated using the “Generate Module Drivers” function. Accordingly, the MOD/PADP/DPAY, RACK and SUBNET blocks should not be inserted manually.
- Driver blocks created by the user must be inserted from the block catalog. Copied "@" blocks will be deleted after the next call of the "Generate Module Drivers" function, since the driver generator does not assign their ID.

The MOD/PADP/DPAY blocks are installed in the acyclic error OBs as well as in OB 1. This ensures that they are also executed following an online download of changes (since OB 100 is no longer executed in this case). The runtime group is assigned a reduction ratio of 16. The blocks are called only every 16th time OB 1 is executed, in order to avoid unnecessary load on the CPU.

Following driver generator startup, the runtime groups in OB 1 is assigned the reduction ratio 16. Changes made later by the user are overwritten. in order to prevent inconsistency during the program cycle.

The system verifies the existence of the PO_UPDAT block in CFC. If it does not exist, it is imported from the library and inserted in the system chart. It is installed in the run sequence at the last position in OB 100. If the block already exists, the system verifies that it is installed in OB 100 after the driver blocks and that it is deleted from all other OBs.

The system verifies the existence of OB_BEGIN/OB_END blocks in CFC. If not found, the system will import the blocks from the library and insert them into the chart and run sequence.

Process image partitions

The CH/PA blocks receive and output their signals via the process image (PI).

The PO_UPDAT block ensures that startup values written to the process image by the CH_AO and CH_DO blocks are transferred to the output modules during a CPU restart and that they take effect immediately.

The OB 1 PI is updated at the cycle control point (no constant scan cycle time). You can achieve a constant scan time by using process image partitions (PIP). PIP update times are configured in HW Config by assigning a PIP to an OB. The PIP update covers the inputs at the start of OB execution and the outputs at the end of OB execution. Process image partitions are module-related, In other words, the fastest signal determines the update of all module signals.

Implementing blocks of new versions

The installation of a **new PCS 7 library** containing modified block types does not initially affect used blocks. Online downloads of changes might not be possible any more.

If you want to update blocks, proceed as follows:

- Delete all @ system charts from the chart folder.

NOTICE
Effects of deletion When the @ system charts are deleted, new DBs are generated during the subsequent generation of module drivers. Since initial values are written in the CPU for one cycle in this process, smooth continuation of work is not always guaranteed.

- Select the menu command **Options > Block Types…**.
- Select the relevant blocks in the “Chart folder” window.
- Click "Clean Up".
  The blocks are deleted.
- Call the “Generate Module Drivers” function in the component view of SIMATIC Manager. New system charts will be created. The driver blocks of the new library will be installed, since they are now no longer available in the CFC. The library must be entered in the “Customize” dialog box.

Additional information

You can find additional information on this in the following sections:

- How to generate module drivers (Page 409)
- Session model for "Generate Module Drivers" (Page 417)
- Interconnection model, module drivers (Page 414)
- Devices supported by the driver generator (Page 411)

17.6 Session model for "Generate Module Drivers"

Configuring the session model for "Generate Module Drivers":

- Configure the hardware and assigns symbolic names to the I/O signals. HW Config will write the signal names directly to the symbol table:
  - Select module.
  - Select the menu command **Edit > Symbols…** in HW Config.

    Note

    If a message appears telling you that an attempt was made to enter an illegal symbolic address, it may be that the size of the process image configured in HW Config is too small. The default size for some CPUs is 512, for example. You can find the settings for the process image in the CPU properties on the "Cycle/Clock Memory" tab. If you insert analog inputs and outputs, their addresses **always** begin at 512 and are thus no longer located in the process image. If you now assign symbolic names, "PIW" or "PQW" will be entered as the address in the symbol table. As a result, the inputs/outputs can no longer be linked with the CH_AI/CH_AO blocks.

    Remedy 1:
    Change the initial addresses of the individual analog modules in the object properties of the module on the "Addresses" tab so that the address range does not exceed the size of the process image.

    Remedy 2:
    If sufficient work memory exists, open the object properties of the CPU, "Cycle/Clock Memory" tab, and change the size of the I/O process image.

- Configure the technological functions in CFC and uses the CH_ blocks (or PA_ blocks) that are interconnected to the signal names taken from the symbol table:
  - Select the menu command **Insert > Interconnection to Address...**.
- In the SIMATIC Manager , call the driver generator with the menu command **Options > Charts > Generate Module Drivers...** to generate, interconnect and configure all module drivers. The following steps are performed automatically:
  - In the first step the function identifies all channel blocks (_AI, _AO, _DI, _DO) used in the CFC charts, and in the next step all I/Os, for example, Value, O_SP, I_OUT_D, that are interconnected to the symbols of the I/O channels configured in HW Config. PA_AO/ PA_DO blocks may require the interconnection of several symbols. Only a "leading" symbol needs to be interconnected in this case, the others are automatically identified and interconnected via the driver generator. By using the symbolic addresses of the symbol table, the corresponding module and channel are identified in hardware configuration data. Based on the module type, the corresponding diagnostic block type (MOD or PADP) is identified and in the system chart a block instance will be created for each I/O module.
  - The parameters required for the diagnostic blocks are obtained from HW Config data and entered in the block instances, as well as in the SUBNET block for optimizing runtime in error OBs.

  - The channel-specific output at the MOD/PADP block is interconnected to the corresponding input at the CH block. If the "value status" module is supported, the address of the value status is obtained and interconnected to the CH/PA block.

Changes made in configuration data (hardware or software) must be processed by the driver generator. Existing driver blocks will not be deleted and recreated, but are rather assigned new parameters. Blocks that are no longer required will be deleted, except for user blocks, and additional blocks are created as required, using blocks imported from the CFC data management instead of the library blocks.

Documenting Programs 18

18.1 Printing charts, dynamic display, I/Os

18.1.1 How to print charts

Printing a chart

To print a chart, select the menu command **Chart > Print**.
The active chart is printed on your default printer.

Options

Print options are available for different printouts. They also apply when printing from the SIMATIC Manager using the menu command **File > Print > Object Content**.

Print what
You can set the following print ranges:

- Current sheet
  Prints the current sheet.
- Current chart partition
  Prints only the overview or sheets or both the overview and sheets of the current chart partition, depending on your selection.
- All chart partitions
  Prints only the overview, the sheets or both the overview and the sheets of all chart partitions of the current chart, depending on your setting.

If the current chart contains chart I/Os, you can select the following print scope:

- Chart with chart I/Os
- Chart without chart I/Os
- Chart I/Os only

If you select print chart I/Os, these will be printed in a table on a separate page. This table contains all relevant information, such as data type, initial value or "Invisible". Chart I/O information (columns) will always be printed, regardless of which columns are visible in the chart I/O window. If there are too many I/Os to fit on one page, additional pages are printed.

The "Overview" and "Sheets" options let you decide whether to print only the overview of the chart, only the sheets or both (depending on the setting for the chart I/Os). These options are disabled when the option "Chart I/Os only" is selected.

Only sheets with contents will be printed. Since at least one chart page will be printed for a chart with or without chart I/Os, an empty sheet will also be printed if the chart consists only of empty sheets.

When you print from the SIMATIC Manager , the lower-level charts, in other words the nested charts of the top chart, are also printed (default). If you do not want to print the lower-level charts, deselect the "Include nested charts (from the SIMATIC Manager)" check box.

Note

The option only applies to the SIMATIC Manager. When printing from CFC, only the open chart will be printed, regardless of this setting. This can also be a nested chart.

Click "Options" in this dialog box to display additional print options. Regardless of the current chart layout, this dialog box lets you customize the print layout for addresses, I/Os and block header information. You change the current layout in the chart with the menu command **Options > Customize > Layout**.

Page setup

With the menu command **Chart > Page Setup...**, you can customize the layout. You specify the paper format for the printer and the page margins.

Print preview

The menu command **Chart > Print Preview** displays a WYSIWYG view of the chart (sheet or overview). Click "Close" or hit <Esc> to return to the standard display.

18.1.2 Creating footers

Introduction

The DOCPRO add-on package can be used for printing a CFC chart, along with footer data. In the footer data, a distinction is made between global data and chart-specific local data.

You can enter global data for the project using DOCPRO or SIMATIC Manager ; local chart-specific data must be input via the CFC editor. Note that chart-specific data will overwrite the global data in the particular chart.

You can also input chart-specific data even if you have not installed the DOCPRO add-on package. Although these data are saved, they will not be available until you have installed DOCPRO.

Keywords in footers

In the global footers, you can enter keywords that will be replaced by current texts in the printout. Below, you can see the available keywords and their meaning:

Keyword	Designation	Meaning	DOCPRO key
$$CN$$	Chart name	Chart name (also nested chart), as entered in the chart properties	$54
$$CC$$	Chart comment	Comment text as entered in the chart properties.	$60
$$A$$	Author	Name as entered in the chart properties	$55
$$DC$$	Date created	Date as entered in the chart properties	$56
$$DM$$	Last modified	Date as entered in the chart properties	$57
---------	---------	---------	---------
$$CH$$	Project path	Path, as entered in the chart properties	
$$PP$$	Storage location of project	Physical storage location, as entered in the chart properties	

In new projects generated under DOCPRO V5.1, you can define the keywords using DOCPRO keywords. This means you do not need to enter the keywords in the footers of the CFC. The footers are, nevertheless, still supported. You still have to enter the texts directly for older projects.

Note

If keywords entered in CFC also specify other print objects (for example, STL blocks or the table of contents of DOCPRO), the printed copy will display the actual keywords, rather than their substitutes, for these objects. This problem no longer exists if you use DOCPRO keys.

When using DOCPRO keys, you must change the standard layout in DOCPRO. Open the "Modify layout" dialog box to replace the default footers with chart-specific footers. You can find information on this in the DOCPRO Online Help or in the manual *DOCPRO: Creating Documentation in Conformance with Standards*.

You must continue to use the keys for "project path" and "Location of the project" (in the lower part of the table) because there are no DOCPRO keys for them.

Footer data for a specific chart

You enter footer data for a specific chart as follows:

1. Select the menu command **Chart > Footers...**.
   A dialog box that contains several tabs ("Part 1" to "Part 4" and "Free Fields") opens.
2. Input corresponding entries in the active fields, for example, the type of document, date when created, document number, data relating to modifications, free texts.

Sheet numbers in the footer

The footer includes a fixed field for continuous page numbering of the print job. For jobs consisting of up to 99 pages, both the consecutive number and the total number of pages is shown, for example, 2/14.

If the job covers more than 99 pages, the total number of pages is not shown but the page number is appended a "+" character. This indicates that there are more pages to come, for example, 1+ to 7+. The number of the last page is appended a "-" character, for example, 127-.

18.1.3 How to print the dynamic display

Procedure

You can print the contents of the (active) dynamic display with the normal print function using the menu command **Chart > Print...**.

The values currently displayed when printing is triggered will be printed.

The column widths are set to optimum values for printing. The printout is in landscape format (default), but can be temporarily changed in the printer setup.

18.1.4 How to print block/chart I/Os

You can print the table of I/Os from the "Properties – Block" and " Properties – Chart" dialog boxes. The "Print..." button is available on the "General" and "I/O" tabs. This button always prints the table of the I/Os.

Procedure

1. Select the “General” or “I/Os” tab.
2. Click "Print…".

The column widths are set to optimum values for printing. The printout is in landscape format (default), but can be temporarily changed in the printer setup.

18.2 Printing chart reference data

Along with the printed charts, the chart reference data provide you with complete documentation of your configuration structure.

You can find more information in the chart reference data online help in the section "What you should know about chart reference data".

18.3 How to display logs

Introduction

The log files do not belong to the complete documentation of the configuration structure. They can, however, provide important information during configuration and commissioning if errors occur or if specific executed operations are to be checked.

Procedure

1. Select the menu command **Options > Logs…**.
   A dialog box with several tabs opens. The tabs are only available when the corresponding functions have been previously executed.
2. Click "Print".
   The messages are printed.

You can select the layout with the "Page Setup" button.

More information on the tabs and buttons is available in the "Help on the log dialog" online help of the dialog box.

Appendix 19

19.1 Data types for S7

19.1.1 ANY, A

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
A	ANY; pointer to data element	only as interconnection to block output or global operand	You will find information on this in the section: Interconnection rules for data types A, DT, S, and ST (Page 160)	

Note

ANY I/Os, for example, SD_1 for SFB 12/BSEND, can also be interconnected with a complete data block (shared address, absolute, for example -> DB1, or symbolic). The complete length of the DB is transferred.

19.1.2 ARRAY

An array consists of a grouped data type with up to 6 dimensions.

CFC does not support this data type.

This means that user-specific blocks that contain this data type will be rejected with an error message during an import to the CFC database.

19.1.3 BLOCK_DB, DB

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
DB	BLOCK_DB; number of a data block (DB)	0... 65535 (number depends on target system)	123	123

19.1.4 BLOCK_FB, FB

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
FB	BLOCK_FB; number of a function block (FB)	0... 65535 (number depends on target system)	123	123

19.1.5 BLOCK_FC, FC

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
FC	BLOCK_FC; number of a function (FC)	0... 65535 (number depends on target system)	123	123

19.1.6 BOOL, BO

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
BO	BOOL; logical number	0-1	0; 1; False; True; F; T	0; 1; 0; 1; 0; 1

19.1.7 BYTE, BY

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
BY	BYTE; sequence of 8 bits	0... 255, (0... FF)	1C; 16#2a; 10#123; 2#10110011	16#1C; 16#2A; 16#7B 16#B3

19.1.8 CHAR, C

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
C	CHAR; individual charac-ter	Depending on the configured Windows character set, "Single byte" or "Multibyte"	"A"; "a"; "0"; "%" A; a; 0; %	"A"; "a"; "0"; "%" "A"; "a"; "0"; "%"

19.1.9 COUNTER, CR

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
CR	COUNTER; number of an S7 counter	0... 65535 (number depends on target system)	123	123

19.1.10 DATE, D

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
D	DATE; date	1990-1-1...2168-12-31	1996-04-29 2000-01-03	96-04-29 00-01-03

19.1.11 DATE AND TIME, DT

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
DT	DATE_AND_TIME or DT; date and time	1990-1-1... 2168-12-31-23:59:59.999	1996-03-16-23:56:19.123	96-03-16:23:56:19

19.1.12 DINT, DI

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
DI	DINT; double integer	-2147483648...2147483647	12345; -17385267; 16#3BC9; 10#123456789; 2#1011010101010101	12345; -17385267; 15305; 123456789 46421

19.1.13 DWORD, DW

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
DW	DWORD; sequence of 32 bits	0... 4294967295, (0... FFFFFFFF)	23AC43BF; 16#1a2b3c4d; 10#1234567890; 2#10010010010010010010	16#23AC43BF; 16#1A2B3C4D 16#499602D2 16#92492

19.1.14 INT, I

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
I	INT; integer	-32768... 32767	4099; -30123; 16#1AC5; 10#12345; 2#0010110010101110	4099; -30123; 6853; 12345; 11438

19.1.15 POINTER, P

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
P	POINTER; pointer to memory area	As interconnection only	- -	- -

Note

When supplying the data type POINTER, certain restrictions and rules must be kept to. Please read the online help of SCL on the POINTER data type. As a general rule, POINTER can be interconnected with any other data type except for POINTER and ANY. If the interconnection is not permitted, you receive a message when you compile the program.

19.1.16 REAL, R

Abb.	Keyword, Type	Value range (normalized) from ... to	Sample entries	Display in the chart
R	REAL; floating-point number	-3.402823e+38... -1.175495e-38... 0.0... 1.175495e-38... 3.402823e+38	22.78; -1234522.456789; -3.456e-3; 2.573e19	2.278e1; -1.23452e+6; -3.456e-3; 2.573e19

Note

STEP 7 maps the data type REAL according to the IEEE standard (to 32-bit short real). Due to the self-restriction, only normalized coding of the numbers is used (see above).

This standard reserves the following additional special bit patterns.

- Not normalized coding with values below the smallest number of the normalized range (-1.175495e-38 < x < 0.0 and 0.0 < x < 1.175495e-38)
- Infinite positive values (+ infinite)
- Infinite negative values (- infinite)
- Illegal bit patterns for numbers, so-called NaN (Not a Number)

When using the data type REAL in CFC, the display/evaluation of online values, for example, dynamic update of values in test mode or when reading back from the CPU, is always based on a calculation with special bit patterns. CFC thus handles those values as follows:

Value	Dynamic display	Activation during readback
Normalized numbers	Floating-point number	Yes
Not normalized numbers	+0.0 or -0.0	No
+ infinite	#+Inf	No
- infinite	#-INF	No
"Not a Number"	#NaN	No

19.1.17 S5TIME, T5

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
T5	S5TIME; duration in S5 format	0h_0m_0s... 2h_46m_30s; 0... 9990 ms in 10 ms intervals, 100 ms... 99900 ms in 100 ms intervals, 1 s... 999 s in 1 s intervals, 10 s... 9990 s in 10 s intervals	1h_30m_0s; 1234567ms; 2h; 32m_5s	1h_30m_0s; 20m_34s_567ms 2h; 32m_5s

19.1.18 STRING, S

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
W	STRING; character string	Depending of the configured Windows character set, "Single byte" or "Multi-byte" (at least 127 characters)	"Batch_127"; This is a longer text.	"Batch_127"; "This is a lon»"

19.1.19 STRING[N], SN

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
SN	STRING[N]; character string	Depending of the configured Windows character set, "Single byte" or "Multibyte" (at least 127 characters)	"Batch_127"; This is a longer text.	"Batch_127"; "This is a lon»"

19.1.20 STRUCT, ST

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
ST	STRUCT;	- -	- -	Name and value of the first elementary data type

19.1.21 TIME, TI

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
TI	TIME; duration	-24d_20h_31m_23s_647ms ... 24d_20h_31m_23s_647ms (-2147483647... 2147483647 ms)	12d_12h_12m_34s_789ms; 123456789ms; 2h; -32m_5s	12d_12h_12m_34s; 3h_25m_45s_678ms 2h; -32m_5s

19.1.22 TIMER, TR

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
TR	TIMER; number of an S7 timer	0... 65535 (number depends on target system)	123	123

19.1.23 TIME OF DAY, T

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
T	TIME_OF_DAY or TOD; time of day	0:0:0.0... 23:59:59.999	12:45:18.012	12:45:18.012

19.1.24 WORD, W

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
W	WORD; sequence of 16 bits	0... 65535, (0... FFFF)	16#bAc1; 10#12345 2#1000011101011010	16#BAC1; 16#3039 16#875A

19.2 References

19.2.1 Synchronize AS-wide Interconnections

Synchronize AS-wide Interconnections

This menu command can be used to determine all one-sided AS-wide interconnections.

- If the interconnection partner is available, the one-sided AS-wide interconnections between these two partners are deleted.
- If one of the interconnection partners is not available, the command has no effect on the interconnections between the two partners and the one-sided AS-wide interconnections are retained.

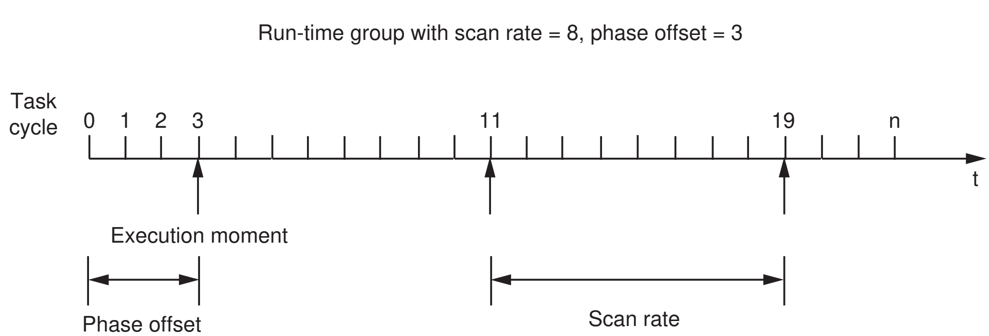
After the command has been executed, a log is displayed showing the deleted and remaining one-sided AS-wide interconnections.

Note

If you have already manually deleted the AS-wide interconnections in the unavailable partner, you nevertheless have to use the command to synchronize the internal time stamp.

19.2.2 Example of reduction ratio and phase offset

The following graphic shows an example for the phase offset and reduction ratio in the clock cycle.



19.2.3 Display of the runtime properties box

The runtime properties box is displayed in the block header with a dark green background (default). Exception: The block that is intended as the "Predecessor for Insertion Position" is displayed with a light green background.

You can find additional information about this in the following sections:

Displaying runtime properties (Page 175)

Display of block execution (Page 198)

19.2.4 "Properties - Block/Chart" dialog box, "General" tab

"Properties - Block/Chart" dialog box, "General" tab

In this tab, you can see the general properties of the:

- Block

or

- Of the nested chart

Block and chart options on the "General" tab

Type

At the block:
Here you can view the type name of the block. If you only see the FB or FC number, you should import the relevant entry with the name of the block into the active symbol table from the symbol table of the source file, or edit the block name for the function block (FB) or function (FC) in the active symbol table.

For the nested chart:
The entry always displays the "Chart" type.

Name

In this box, you can view and edit the name of the block instance or chart. The maximum string length for blocks is 16 characters, for nested charts 22 characters. The name may not include the following characters: \ / . " %.

Note

Please observe the following information:

- Remember when assigning names that the variable name must not be longer than 128 characters for transfer to the OS ("Compile OS"). The name consists of the following elements:
  - The name of the folder in the hierarchy path
  - The chart name
  - The block name
  - The separator (period)
  - The I/O name (variable name)
- When transferring to the OS, in addition to the illegal characters listed above, the following characters are not permitted in the name and are automatically replaced by the **$** character: Space **? * ' :**

Tip: Never use any of the characters listed above when selecting names.

Comment

In this box, you can view and edit the block or chart comment. You can enter a maximum of 80 characters. Only the first 14 characters (large block and nested chart) or 7 characters (small block) are displayed in the header.

The comment will only be adopted by WinCC, if it was edited at this instance prior to OS compilation.

Group (only for blocks with messages)

In this field you can assign the block to a group that is intended for hiding messages. The name of the group can have a maximum of 24 characters.

Options on the "General" tab for blocks

The following options only exist for the block properties.

Inputs

Here, you can see the number of block inputs of the selected block. If the block has a variable number of inputs and the same data type, for example, NAND, OR, etc., you can change the number of block inputs with the menu command **Edit > Number of I/Os**.

Internal identifier (S7 only)

When you implement block types as functions or function blocks, you will see the block type (for example, FC or FB) and number. The block folder in SIMATIC Manager displays this internal block identifier as "Object name".

Instance DB (S7 only)

The program displays the DB number of blocks mapped to a data block.

Name (header)

Displays the name of the block as specified in the internal block header at the time the block was created.

Family

Displays the family name of the block as specified in the internal block header at the time the block was created. This is used, for example, as a sorting criterion in block catalogs.

Author

Displays the name as specified in the internal block header at the time the block was created (for example, with basic operations "CFC-BOP" or technological blocks "TECH").

To be inserted in OB/tasks

There are blocks that must be installed in certain tasks, for example, to ensure that the startup behavior, process-control message capability and other properties will take effect. The list displays the tasks where CFC has automatically installed the block. You can edit the run sequence later with the menu command **Edit > Run Sequence...**.

"OCM possible" area

Here you can define the response of a block that can output messages or can be controlled and monitored from an OS.

The option "OCM possible" does not influence message transfer to the OS.

- "OCM possible" check box
  Select or deselect the "OCM possible" check box to decide whether this block is entered in the watch list on the OS for operator control and monitoring. When you deselect the check box, all elements of this group are switched to inactive. They are then no longer relevant for execution but they retain their setting.
- "Operator Control and Monitoring..." button
  Click "Operator Control and Monitoring" to open the dialog box where you can set special properties. There is usually no need to call this dialog box when using WinCC and the PCS 7 library blocks, since the properties are set here by default.
- "Create block icon" check box
  When you select/deselect the "Create block icon" check box, you decide whether or not a block icon is created for this block.
  The block icon can be activated for instances of SFC types. The variants of the block icons are configured in the object properties of the SFC type, however.

- Input field
  In the "Block icon" input field, you can specify the icon WinCC displays for this block. You can thus set different variants for the same block type, if they exist.
  A maximum of 16 characters is possible.
  The templates of the block icons and their variants are located in the picture "@PCS7TypicalsBasisLibraryV8.pdl".
  The variants of a block icon can be distinguished by the string "/<number>" at the end of the icon name, for example, <name>/2".

Variant	Remark
1	Standard style PCS 7
2	New style based on the APL
<Variant name>	User-defined variants; user-defined variant name as the name of a block icon; max. 16 characters

  The input field is active if the "Create block icon" check box is selected. The input field is disabled if you clear the "Create block icon" check box. However, the entry is retained.
  You can find more information on block icons in the section "Operator control and monitoring (Page 170)" or in the section "Configuring SFC block icons" of the "SFC Visualization" online help or documentation.
- When you select/deselect the "MES-relevant" check box, you decide whether or not the information of this block is transferred to the enterprise levels MIS/MES when requested. This input field is only enabled if you have selected the "OCM possible" check box.

"Special properties" area

- Click "Messages..." to open a dialog box where you can edit the block's message texts.
- If the "Read-back enabled" check box is selected, the block is included in the read back (default). The block has the system attribute S7_read_back = 'true'.
  Depending on the scope set in the Read Back dialog box, all block I/Os are read back or only those with the "Operator control and monitoring" system attribute (S7_m_c := 'true') and/or those with a special designation (S7_read_back := 'true').
  You can also exclude the entire block from the read back if you deselect the check box (S7_read_back := 'false').
- Use the "Technolog. assignments..." button to open a dialog box in which all assigned objects are listed.
  Select an object and confirm it with the "Go to" button to jump directly to this object. This button only exists if there are assignments.

"OK" button

This button applies all the changes you have made in the dialog box. The dialog box closes.

"Apply values" button

This button is active only when you are in test mode and have changed one or more values in the "Value" column of the "I/Os" tab.

The button applies only the modified values in the "Value" column on the CPU. The dialog box remains open so that you can change a value more than once and follow the result in the block in the chart.

"Print" button

This button enables you to print the table of input/outputs. The tab switches from "General" to "I/Os" when you click the button.

The table is printed in landscape format. The columns are set to optimum width and then set back to their previously set width.

"Cancel" button

This button discards all unsaved changes and closes the dialog box. Modified values that you have already accepted with the "Apply values" button are retained.

19.2.5 "Properties - Block/Chart" dialog box, "I/Os" tab

"Properties - Block/Chart" dialog box, "I/Os" tab

The table on this tab displays the I/Os of the selected object.

All existing elements are listed in table columns, regardless of the object type (block instance, nested chart, external view of the SFC chart, instance of an SFC type). However, not all columns are globally relevant to every object type and you may not be able to change them. You can never edit grayed out fields.

Meanings of table columns

#

This column is always visible. I/Os are assigned numbers in this column. After you open the object properties dialog, you will see the consecutive numbering in ascending order. Sorting the I/Os changes the order. The original (ascending) order is restored after you exit the dialog box.

Name

Here, you see the name of the I/O. This column is always visible.

I/O

Here, you can see the type of I/O (IN = Input, OUT = Output, IN_OUT = In/Out).

Type

Here, you see the data type of the I/O.
You can find detailed information on this in the section: Data types for S7 (Page 159)

Value

Here, you can view and edit I/O parameter values. Different ranges of values and inputs are permitted, depending on the data type. You cannot modify grayed out values:

- At interconnected I/Os and an empty field (contains no value)
- Of I/Os that are not configurable (system attribute: S7_pram := 'false')
- Of a textual interconnection
  The value (of the block type) is used here as a substitute value for the interconnection partner.

Instead of an absolute numeric value, a value identifier can also be displayed here if value identifiers were configured for this I/O and the "Value identifier" option is enabled in the settings for the display. In this case, you can select the text to be displayed in a combo box, for example, the display text of an enumeration. The button for the drop-down list is displayed when you click in the combo box.

Interconnection

This shows the connected partner when I/Os are interconnected. When an output has several interconnections, all inputs connected to this output are shown.

Background:
A maximum of 160 I/Os per interface side (left side IN and IN_OUT, right side OUT) can be displayed in the CFC. When there are more than 160 I/Os per interface side they are switched to invisible. This occurs from the last I/O onward. This may include interconnected I/Os. Although these interconnections are no longer visible, they can be recognized only in the "Interconnection" column.

Note: If the I/O of one interconnection partner is invisible, the interconnection of the visible I/O is led to the sheet bar and the target of the interconnection is identified by "(INVISIBLE)".
If both partners in the interconnection are switched to invisible, the interconnection can no longer be recognized in the chart. This also applies to textual and symbolic interconnections and for interconnections to runtime groups.

If an I/O cannot be interconnected (S7_link := 'false'), the text <cannot be interconnected> is entered.

Add forcing

Displays the selected I/O at which the add forcing function is enabled. This option is only available if forcing has been globally enabled in the SIMATIC Manager .

Forcing active

Displays the selected I/O at which the forcing active function is set. This option is only available if "Add forcing" is active.

Force value

Displays the force value. You can only edit this value if "Add forcing" is active. This value depends on the data type of the I/O.

SFC access

Shows the I/O which are selected for read / write access by an SFC. This identifier is of particular significance to hidden I/O, because this SFC access is not shown in the block representation of the chart.

Technological assignments

Here the connections are selected that are connections of a control module. You copy and insert the connections with a shortcut menu.

Comment

Here you can view and edit the I/O comment (maximum of 80 characters).

The comment of an interconnected input can be an input comment (I/O comment) or an interconnection partner comment (interconnection comment). The display depends on the settings made in "Customize Display". Open the dialog box using the menu command **Options > Customize > Layout....** If the interconnection comments are to be displayed in the CFC chart, select the "Interconnection comment" check box in the "Parameter" group of the dialog box.

You cannot modify the interconnection comment in the input object properties.

Invisible

Here, you can decide whether interconnected or not interconnected I/Os are to be displayed or hidden.Check mark = hiddenNo check mark = visible.

If the I/O of one interconnection partner is invisible, the interconnection of the visible I/O is led to the sheet bar and the target of the interconnection is identified by "INVISIBLE".

If both interconnection partners or the I/O with a symbolic or textual interconnection is switched to invisible, this interconnection can no longer be recognized in the chart, but only in the "Interconnection" column. The same applies to interconnections to runtime groups.

Note

If the block contains invisible interconnected I/Os, a colored triangle appears in the top right corner of the block header. This also applies to SFC access.

Watched

Here you can set the I/Os so that they are logged in for test mode. This allows you to view and edit the actual values of the CPU.

Archive

Here, you can mark I/Os of the data type BOOL, BYTE, WORD, DWORD, INT, DINT, and REAL for the following types of archiving:

- No archiving
- Archiving
- Long-term archiving

Requirement: The I/Os are intended for operator control and monitoring.

The boxes in this column can only be edited when the I/O is intended for operator control and monitoring (system attribute S7_m_c := 'true'). This applies regardless of whether the current block is switched to operator control and monitoring.
With the integrated drop-down list, you can specify that the I/O is not relevant for archiving ("No Archiving"), or is marked for "Archiving", or for "Long-term archiving".

You can find additional information on this in the following section: Configuring archive tags (Page 172)

Identifier

Here you can view the identifier of I/Os that are not of the type BOOL and you can enter identifiers with a maximum string length of 16 characters, for example, "Setpoint" or "CP". The OS uses these texts for displays and logs.

Unit

Here, you can select the most common units from a drop-down list for I/Os that are not of the BOOL data type. The button for the drop-down list is displayed when you click in the combo box.

Note

The set of units is installed with CFC and can be modified or supplemented in the SIMATIC Manager . You can find additional information on this in the following section: Configuring Shared Declarations (Page 163).

Text 0

Here, you can enter a text with a maximum length of 16 characters for the value "0" for I/Os of the BOOL data type, for example, "Close" and "Closed".
In the "Value" column you can select this text or "Text 1".

Note

If the I/O is also assigned the system attribute "S7_enum", this box is deactivated. In this case, only the enumeration is used as value identifier.

Text 1

Here, you can view and enter a text with a maximum length of 16 characters for the value "1" for I/Os of the BOOL data type, for example, "Open" and "Opened".
In the "Value" column you can select this text or "Text 0".

Note

If the I/O is also assigned the system attribute "S7_enum", this box is deactivated. In this case, only the enumeration is used as value identifier.

Enumeration

If this involves an I/O that is assigned the "S7_enum" system attribute, you can select the enumeration in the drop-down list created in the current project in "Shared Declarations" for I/Os of the data type BOOL, BYTE, INT, DINT, WORD, DWORD.

The display names that you can select from the "Value" box in the drop-down list are assigned as values to each enumeration if the "Value identifier" option is activated in the "Customize Display".

You can select the empty field in the "Enumeration" drop-down list to delete an enumeration. The numerical value is then displayed in the "Value" box instead of the display name.

Note: When the enumeration is deleted, the "S7_enum" system attribute is retained at the block I/O, in other words an empty string is entered instead of the enumeration previously shown.

You can find additional information on the enumerations in the section: Configuring Shared Declarations (Page 163)

Parameters

You can here identify the I/Os you want to display on the "Parameters" tab of the process object view. You can change this setting in the SIMATIC Manager using the menu command **Options > Process Objects > Select I/Os...**.

Signal

Here you can identify the I/Os you want to display on the "Signal" tab of the process object view. You can change this setting in the SIMATIC Manager using the menu command **Options > Process Objects > Select I/Os...**.

Options are: "Parameter" or "Signal", or none at all.

MES-relevant

When the "MES-relevant" check box is selected, this acts as a filter so that if an MES system is connected, only the control system information of interest for MES is transferred. With this option, you can mark individual I/Os with OCM capability as being relevant or irrelevant for MIS/MES. You can identify these I/Os as relevant regardless of the setting of the "MES-relevant" check box on the "General" tab.

Operator authorization level

This option is only available if the "Op_Level" attribute is available for the block type. You can set a value between 1 and 99 here.

OS additional text

Here you can enter a label text for the button in the faceplate. You can use this button to jump to the faceplate of the interconnected block. The text can be edited.

The input field is only active if the block has screen jump functionality in WinCC.

Sorting the columns

You can sort the I/Os in this table. This sorting order is temporary and restored to the original order when you close the dialog box.

Click a column title to sort all I/Os. Each click toggles the alphabetical order: ascending/descending. The sorting order of columns with selection boxes is determined by: not set/set. The "#" column indicates changes to the sorting order.

Buttons

"OK" button

This button applies all the changes you have made in the dialog box and closes the dialog box.

"Apply values" button

This button is active only when you are in test mode and have changed one or more values in the "Value" column of the "I/Os" tab.

The button applies only the modified values in the "Value" column on the CPU. The dialog box remains open so that you can change a value more than once and follow the result in the block in the chart.

"Print" button

This button enables you to print the table of input/outputs. The tables are printed in landscape format. The columns are set to optimum width and then set back to their previously set width.

"Cancel" button

This button discards all unsaved changes and closes the dialog box. Modified values that you have already accepted with the "Apply values" button are retained.

19.2.6 "Properties - Block/Chart" dialog box, "Type update settings" tab

"Properties - Block/Chart" dialog box, "Type update settings" tab

The table on this tab displays the I/Os of the selected object.

In this tab, you can select various attributes of a parameter to exclude them from type-instance synchronization (type update). Selected attributes will not be updated while synchronizing type and instances.

Note

Some parameters are implicitly excluded from type-instance synchronization. For more information, refer to the section "Effect of changes to the type/instance synchronization with control modules (Page 225)".

Meanings of table columns

#

This column is always visible. I/Os are assigned numbers in this column. After you open the object properties dialog, you will see the consecutive numbering in ascending order. Sorting the I/Os changes the order. The original (ascending) order is restored after you exit the dialog box.

Name

Here, you see the name of the I/O. This column is always visible.

I/O

Here, you can see the type of I/O (IN = Input, OUT = Output, IN_OUT = In/Out).

Type

Here, you see the data type of the I/O.
You can find detailed information on this in the section: Data types for S7 (Page 159)

If the following options are selected for a parameter, the corresponding attribute of the parameter is excluded from type update explicitly:

- **Ignore value**
  Value of the parameter is ignored for type update and the parameter is marked in “Pink” color in the block I/O (generally referred to as ”Pink” parameters).
- **Ignore comment**
  Comment text for the parameter is ignored for type update.
- **Ignore Text0**
  Text for the parameter value 0 (system attribute S7_string_0) is ignored for type update.
- **Ignore Text1**
  Text for the parameter value 1 (system attribute S7_string_1) is ignored for type update.
- **Ignore identifier**
  Identifier (system attribute S7_shortcut) is ignored for type update.
- **Ignore unit**
  Measuring unit of the parameter (system attribute S7_unit) is ignored for type update.
- **Ignore enumeration**
  Enumeration (system attribute S7_enum) is ignored for type update.
- **Ignore parameter/signal**
  Setting for displaying parameter/signal in the "Process Object View" (system attribute S7_edit) is ignored for type update.

Sorting the columns

You can sort the I/Os in this table. This sorting order is temporary and restored to the original order when you close the dialog box.

Click a column title to sort all I/Os. Each click toggles the alphabetical order: ascending/ descending. The sorting order of columns with selection boxes is determined by: not set/set. The "#" column indicates changes to the sorting order.

Buttons

"OK" button

This button applies all the changes you have made in the dialog box and closes the dialog box.

"Print" button

This button enables you to print the table of input/outputs. The tables are printed in landscape format. The columns are set to optimum width and then set back to their previously set width.

"Cancel" button

This button discards all unsaved changes and closes the dialog box. Modified values that you have already accepted with the "Apply values" button are retained.

19.2.7 Enable Attribute

Function

The enable attribute enables or disables a runtime group (on = 1, off = 0) and represents a master control. As long as “0” is set the runtime group will not be executed, regardless of any other conditions.

Enabling/disabling runtime groups dynamically

The default setting of the enable attribute is 1. It can also be set dynamically, i.e., the output value of a CFC block can enable or disable the runtime group. You can interconnect the binary output of a block to the runtime group.

Enabling/disabling runtime groups in the runtime editor

You can enable/disable runtime groups via the runtime editor window.

To enable a runtime group, select the “Active” check box in the object properties of the corresponding runtime group. To disable a runtime group, deselect the “Active” check box in the object properties of the corresponding runtime group. You can also do this via the shortcut menu commands **Switch on Runtime Group** and **Switch off Runtime Group**. This avoids you having to open the object properties dialog box.

Note

If a runtime group is interconnected to a binary output of a block, this output value always controls the runtime group. In this case, the “Active” option is ignored.

19.2.8 Displaying tooltips using the cursor

In test mode you can display online values as tooltips by placing the cursor on an I/O or an interconnection line. Here, the cursor functions like the "test prod" of a measuring device.

Requirement: It must be possible to update the I/O online (not an unconnected FC input, for example).

The following applies in respect of tooltip display:

- The information is actively retrieved from the CPU by placing the cursor on an I/O or the interconnection even if the I/O has not been registered for monitoring. Updating is performed once a second, regardless of the selected watch cycle.
- The value is substituted with the online value and the background color of the tooltip changes from light yellow to yellow, as it does with watched I/Os. With I/Os that cannot be operator controlled and monitored, for example, when they are interconnected, the background is yellow-gray.
- The output value is displayed for an interconnection. Exception: With interconnections to shared addresses, the input value is displayed. Inversions are taken into account in this case.

19.2.9 Searching for objects in the catalog

Searching in the catalog

You can specify a text in the input field of the catalog and search for it by clicking this button:

If the text you entered is not found as an object name, CFC searches for a block with a corresponding comment. The folder, for example, of the block family, or the library in which the block is located is opened and the block is selected.

The block is searched for starting from the selected object (library, folder or block) in the active catalog window. All the libraries and folders are searched until the required object is found or until the search returns to the selected object.

With the "Find initial letter" check box you can decide whether the search starts at the initial letters (restricted search) or whether any part of a name or comment should be searched for (free search; default).

During the search, a dialog box with a progress display is shown. Here, you can stop the search if it takes too long, for example, when you first search through libraries.

The search stops when an object with the specified letters is found.

By clicking on the button mentioned above, you can search for other objects with these letters. The search is completed when it returns to its starting point.

19.2.10 Phase offset

Function

Phase offset allows a uniform distribution of load within the CPU. It must be considered in conjunction with "n", the reduction ratio. The group is executed as often as specified by "n", in each case offset by "m" units of the task cycle. "m" is an integer, where **0 <= m <= (n-1)**.

Default: 0, no phase offset

Additional information

You can find additional information on this in the following sections:

Reduction ratio (Page 450)

Example of reduction ratio and phase offset (Page 432)

19.2.11 Signal processing with driver concept up to V5.2

The driver and message concept described here applies if you want to use or continue to use CH blocks with version < 2.0.

Driver and message concept

For this functionality, block types are used to isolate hardware data from software configuration data:

- The user interconnects channel-specific blocks (CH blocks) of the CFC chart and assigns corresponding signal names from the symbol table.
  The CH block is used for signal preprocessing and can be configured and interconnected independent of the hardware. It is thus an object of the system functions and is always processed along with those functions.
  Four different types are available for signal preprocessing:
  - Signal preprocessing of analog input values (CH_AI)
  - Signal preprocessing of analog output values (CH_AO)
  - Signal preprocessing of digital input values (CH_DI)
  - Signal preprocessing of digital output values (CH_DO)

  For additional information on the functions and operating principle of CH blocks, refer to the context-sensitive help (<F1>) for the block.
- The system automatically generates the multi-channel blocks (MOD blocks) for the module-specific functions, that is, it imports the blocks from a block library and inserts these into a system chart, programs and interconnects them.
  The MOD block is also known as diagnostics block. It handles all special situations such as startup and error events, for example, by providing process control messages and the signal value status (1 = good, 0 = bad).
- In addition to the diagnostic blocks and in order to ensure that not all MOD blocks report a rack failure if a rack fails, a RACK block is automatically added for each rack.
  The RACK block is responsible for signaling in this case. The MOD blocks recognize in OB 86 that the event has been processed and do not report.
- The SUBNET block controls the runtime groups for each configured DP chain and is used to reduce acyclic OB execution times. When an event occurs, only the blocks that are actually affected will be called. The SUBNET block is also installed automatically.
- The PO_UPDAT block ensures that the start values written to the process image by the CH_AO and CH_DO blocks are transferred to the output modules during a CPU restart. This means that these values are effective immediately when the CPU changes to RUN.
- The MSG_CSF message block is implemented in PCS 7 process control systems in order to prevent errors, for example, rack failure, module failure, from triggering a CPU transition to STOP, In other words, it is implemented by installing the MSG_CSF block in special OBs generated by the system.

For additional information on the functions, functionality and message capability of RACK and SUBNET blocks, refer to the context-sensitive help (<F1>) for the relevant block.

How the "Generate Module Drivers" function works

A system chart is generated automatically and the MOD, RACK, SUBNET blocks and the process control message block MSG_CSF and the PO_UPDATE block for outputting the process image are inserted. The system chart is assigned the name "@1" and can contain up to 52 blocks, with 2 blocks per chart partition. If additional blocks are generated, another system chart will be created and assigned the name "@2".

The blocks are imported from the libraries when you initially generate the module drivers and if the CFC database does not yet contain the corresponding driver blocks or MSG_CSF. The block search starts at the path defined in the "Settings – Generate Block Drivers" dialog box. If a block is not found, it searches through the default paths specified during setup of the PCS 7 libraries.

Note

This method prevents new versions of block types being used in the program if blocks with the same name already exist in the CFC data management. It also prevents global block type changes for these blocks.

The MOD blocks for each rack will be installed in runtime groups, including the RACK block. Those blocks and the runtime groups are identified in the run sequence by the "@" prefix.

Note

Please observe the following information:

- Objects identified by the "@" character should not be modified by the user, but should only be manipulated using the “Generate Module Drivers” function. The same applies to manual insertion of MOD, RACK and SUBNET blocks.
- Driver blocks created by the user must be inserted from the block catalog. When "@" blocks are copied, they will be deleted the next time the "Generate Module Drivers" function is executed.

The MOD blocks are installed in the acyclic error OBs as well as in OB 1. This ensures that they are also executed following an online block download, since OB 100 is no longer executed in this case. The runtime group is assigned a reduction ratio of 16. The blocks are called only every 16th time OB 1 is executed, in order to avoid unnecessary load on the CPU.

The system verifies the existence of the PO_UPDAT block in CFC. If it does not exist, it is imported from the library and inserted in the system chart. It is installed in the run sequence at the last position in OB 100. If the block already exists, the system verifies that it is installed in OB 100 after the driver blocks and that it is deleted from all other OBs.

The system checks whether an MSG_CSF exists in CFC. If this is not the case, it will be imported from the "PCS 7 Technology\Blocks" library, inserted in the system chart and in the run sequence of the cyclic interrupt OB 32 and in the following startup/special OBs: OB 72, OB 81, OB 83, OB 84, OB 85, OB 86, OB 87, OB 100, OB 121, and OB 122.
No warning is issued when there is more than one MSG_CSF.

Process image partitions

The CH blocks receive and output their signals via the process image (PI).

The PO_UPDAT block ensures that startup values written to the process image by the CH_AO and CH_DO blocks are transferred to the output modules during a CPU restart and that they take effect immediately.

The OB 1 PI is updated at the cycle control point (no constant scan cycle time). You can achieve a constant scan time by using process image partitions (PIP). PIP update times are configured in HW Config by assigning a PIP to an OB. The PIP update covers the inputs at the start of OB execution and the outputs at the end of OB execution. Process image partitions are module-related, In other words, the fastest signal determines the update of all module signals.

Implementing blocks of new versions

The installation of a new PCS 7 library containing modified block types initially has no effect on blocks used already.

Proceed as follows to update the blocks:

1. Delete all @ system charts from the chart folder.
2. Select the menu command **Options > Block Types…**.
3. Select the relevant blocks and click "OK".
   The blocks are deleted in the "Chart folder" window.
4. Call the "Generate Module Drivers" function in the component view of SIMATIC Managers .
   New system charts will be created. The driver blocks of the new library will be installed, since they are now no longer available in the CFC. Enter the library in the “Customize” dialog box. If MSG_CSF is not found in this library, it is searched for in the default path that was entered during setup.

Model session, version 1

- Configure the hardware and assigns symbolic names to the I/O signals. The signal names are entered directly in the symbol table by HW Config:
  Select the module
  Select the menu command **Edit > Symbols...** in HW Config.
- Configure the technological functions in CFC using CH blocks that must be interconnected under the signal names taken from the symbol table.
  Select the menu command **Insert > Interconnection to Address...**.
- In theSIMATIC Manager , call the driver generator with the menu command **Options > Charts > Generate Module Drivers...** to generate, interconnect and configure all module drivers. The following steps are performed automatically:
  - The system identifies all channel blocks (CH_AI, CH_AO, CH_DI, CH_DO, CH_CNT, CH_CNT1, CH_MS) of the CFC charts in the first step, and all symbols (name and address) interconnected to the "Value" I/Os in the next. Using the symbolic addresses of the symbol table, the system identifies the corresponding module and channel in hardware configuration data. Based on the module type, it determines the corresponding MOD block type and generates a block instance for each I/O module in the system chart.
  - Parameters required for MOD blocks are fetched from hardware configuration data and entered at the block instances and at the SUBNET block that is used for optimizing runtime in the error OBs.
  - The channel-specific output at the MOD block is interconnected to the corresponding input at the CH block. If the "value status" module is supported, the address of the value status is obtained and interconnected to the CH block.

Changes made in configuration data (hardware or software) must be computed via the "Generate Module Drivers" function. Existing driver blocks will not be deleted and recreated, but are rather assigned new parameters. Blocks that are no longer required will be deleted (except for user blocks) and additional blocks will be created as required. using blocks imported from the CFC data management instead of the library blocks.

Model session, version 2

- Configure the technological functions in CFC using CH blocks that must be interconnected under the signal names taken from the symbol table.
  Select the menu command **Insert > Interconnection to Address...**.
  Entering the signal name in the dialog box
- Configure the hardware and assigns symbolic names to the I/O signals. The signal names are entered directly in the symbol table by HW Config:
  Select the module
  Select the menu command **Edit > Symbols...** in HW Config.

Tip on configuring the hardware:

If you have configured the software first and already interconnected the block I/Os to the symbolic names of the I/O signals with the menu command **Interconnection to Address...** and then want to configure the hardware, you can copy the symbolic names from the "Cross-References of Addresses" list and paste them into HW Config data.

19.2.12 Reduction ratio

Function

The reduction ratio specifies whether the runtime group is processed each time the OB is executed or only every nth time the OB is executed. "n" is an integer (n = 2t, where 0 <= t <= 15). The run cycles are a multiple of the time base.

Default: 1, the runtime group is executed in every run cycle.

Example:
Basic cycle of a cyclic interrupt: 1 sec
Possible clock cycles with reduction ratio: 2, 4, 8, 16, etc.

Additional information

You can find additional information on this in the following sections:

Phase offset (Page 445)

Example of reduction ratio and phase offset (Page 432)

19.2.13 "Properties - Input/Output" dialog box

"Properties - Input/Output" dialog box

In this dialog box you can edit the I/O of a block, nested chart or SFC chart you have selected from the chart.

Note

The elements described at this point may not be relevant to the above mentioned object types or may not exist at all. Grayed out boxes cannot be edited.

Options in the "Properties - Input/Output" dialog box

Block or **Chart**

Here you can view the following information about the selected I/O:

- The name of the block type and the block name following the separator or
- The name of the nested chart or SFC chart

I/O

Here you can view the following information about the current I/O:

- The I/O name
- The I/O type (IN=input, OUT=output, INOUT=in/out parameter)
- The data type (in brackets)

Value

Here you can edit the value of the selected I/O. The I/O should not be interconnected and must be configurable.
Different value ranges and inputs are permitted, depending on the data type.

Exception: In a textual interconnection, this value (of the block type) is used as substitute value for a real interconnection partner. The field is grayed out, in other words, the value cannot be edited.

The "Value" box contains a drop-down list with texts if the I/O is an I/O with value identifier and the "Value identifier" option is activated in the "Customize Display".

You can enable and disable the value identifier using the menu command **Options > Customize > Layout...** in the "Customize Display" dialog box.

Note

If you have more than 8 characters of text, you can define the characters to be displayed. You do this by including the "=" character in the text. If there is an equal sign in the text, the first 8 characters to its right are displayed. If there is no equal sign in the text, the first 8 characters are displayed.

You cannot change the value of the following I/Os:

- Interconnected I/Os
- I/Os with the "Not configurable" attribute (e.g., system inputs such as event ID)
- I/Os of the data type "ANY" (since the "ANY" data type is only determined by connecting it to an I/O of a defined data type)

Inverted

Interconnected inputs of the "BOOL" data type can be inverted via the "Inverted" radio button.

Invisible

With the "Invisible" check box, you can hide connected or unconnected I/Os in the chart display.

If only the I/O of one interconnection partner is invisible, the interconnection of the visible I/O is led to the sheet bar and the target of the interconnection is identified by "(INVISIBLE)".

If both interconnection partners or the I/O with a symbolic or textual interconnection is switched to invisible, this interconnection can no longer be recognized in the chart but only in the object properties of the block in the "Interconnection" column of the "I/Os" tab. The same applies to interconnections to runtime groups.

Note

If the block contains invisible interconnected I/Os, a colored triangle appears in the top right corner of the block header.

Watched

You can use the "Watched" check box to set the I/O so that it displays the actual values of the CPU during test mode.

Text for state 0

This option is only visible when you are editing an I/O of the data type "BOOL" whose system attribute is set to "S7_string_0". You can enter a maximum of 16 characters for the "0" value. The OS uses these texts for displays and logs, for example, "Close" and "Closed".
In the "Value" box, you can select this text or "Text 1".

Text for state 1

This option is only visible when you are editing an I/O of the data type "BOOL" whose system attribute is set to "S7_string_1". You can enter a maximum of 16 characters for the "1" value. The OS uses these texts for displays and logs, for example, "Open" and "Opened".
In the "Value" box, you can select this text or the one from "Text 0".

Enumeration

This option is visible only when the I/O has the system attribute "S7_enum" assigned to it. From the drop-down list, you can select one of the enumerations created in the current project in "Shared Declarations" for I/Os of the data type BOOL, BYTE, WORD, DWORD, INT, DINT.

The display names that you can select from the "Value" box in the drop-down list are assigned as values to each enumeration if the "Value identifier" option is activated in the "Customize Display". You can find additional information on the enumerations in the section: Configuring Shared Declarations (Page 163)

You can select the empty field in the "Enumeration" drop-down list to delete an enumeration. The numerical value is then displayed in the "Value" box instead of the display name.

Note: When the enumeration is deleted, the "S7_enum" system attribute is retained at the block I/O, in other words an empty string is entered instead of the enumeration previously shown.

Comment

The comment of an interconnected input can be an input comment (I/O comment) or an interconnection partner comment (interconnection comment). The display depends on the settings made in "Customize Display". Open the dialog box using the menu command **Options > Customize > Layout...**.

If the interconnection comments are to be displayed, select the "Interconnection comment" check box in the "Parameter" group of the dialog box. You cannot modify the interconnection comment at the input.

If the comment is not an interconnection comment, you can enter a comment with a maximum of 80 characters for the I/O here. If you have selected the "Comment" option in the “I/Os” group with the menu command **Options > Customize > Layout**, the first 12 characters (wide block display) or 8 characters (narrow block display) of the I/O comment are displayed in the chart instead of the I/O name.

Operator authorization level

This option is only available if the "Op_Level" attribute is available for the block type. You can set a value between 1 and 99 here.

Identifier

This option is only visible when you are editing an I/O of a data type other than "BOOL" whose system attribute is set to "S7_shortcut". The identifier you enter must not exceed a maximum string length of 16 characters. The OS uses these texts for displays and logs, for example, "Setpoint" and "CP".

Unit

This option is only available when you are editing an I/O of a data type other than "BOOL" and have assigned the system attribute "S7_unit" to it.

Here, you can select the most common units from a drop-down list for the I/O. The button for the drop-down list is displayed when you select the combo box.

Note

The set of units is installed with CFC and can be modified or supplemented in the SIMATIC Manager . You can find additional information on this in the following section: Configuring Shared Declarations (Page 163)

Archive

This option is only available when the I/O is intended for operator control and monitoring (system attribute S7_m_c := 'true').
You can specify the following with this drop-down list:

- If the I/O is no longer relevant for archiving: "No Archiving" (S7_archive := 'false')
- If the I/O is to be identified for "Archiving" (S7_archive := ‘shortterm’)
- If the I/O is to be identified for "Long-term archiving" (S7_archive := ‘longterm’)

The I/Os identified as being relevant for archiving are created as archive tags during compilation of the OS in the OS project and, if it does not already exist, a process value archive is also created. If an I/O relevant for archiving is deleted later or is changed to "No Archiving", the corresponding archive tag is deleted the next time you compile the OS.

You can find additional information on this in the following section: Configuring archive tags (Page 172)

OS additional text

Here you can enter a label text for the button in the faceplate. You can use this button to jump to the faceplate of the interconnected block. The text can be edited.

The input field is only active if the block has screen jump functionality in WinCC.

"Technolog. assignments..." button

Use the "Technolog. assignments..." button to open a dialog box in which all assigned objects are listed. Select an object and confirm with the "Go to" button to jump directly to this object. This button only exists if there are assignments.

Forcing

These options are only available if the global "Support forcing" option has been set in the component view of the SIMATIC Manager (dialog box "**Chart folder properties"**, **tab: Advanced**).

- **Add forcing**
  You can enable the add forcing function for the I/O by setting this option. Chart inputs/ outputs cannot be logged on for forcing.
- **Forcing active**
  You can enable forcing for the I/O by setting this option. This option is only available after the "Add forcing" function was selected.
- **Force value**
  You can enter a force value for the I/O in this field. This value to be entered depends on the data type of the I/O. This option is only available after the "Add forcing" function was selected.

MES-relevant

This option is only available when the I/O is intended for operator control and monitoring (system attribute S7_m_c := 'true') and the "OCM possible" option is activated in the object properties of the block.

When the "MES-relevant" option is set, this acts as a filter so that if an MES system is connected, only the control system information of interest for MES is transferred.

Process object view

- **Parameter:**
  Here you mark the I/O if you want to set parameters or interconnect it with another I/O. It is then displayed on the "Parameter" tab of the process object view. You can change this setting in the SIMATIC Manager using the menu command **Options > Process Objects > Select I/Os...**.
- **Signal:**
  Here you mark the I/O if you want to interconnect it with a signal in the process object view. It is then displayed on the "Signal" tab. You can change this setting in the SIMATIC Manager using the menu command **Options > Process Objects > Select I/Os...**.

Options are: "Parameter" or "Signal", or none at all.

SFC Access...

This button is only available if SFC access exists on this I/O. This access is indicated by colored markers at the I/O in the CFC chart:

- Write access = marker below the I/O
- Read access = marker above the I/O

Use this button to open a dialog box that contains a list of current SFC accesses. Double-click the chart name to open the selected SFC chart. The accessing chart element is highlighted.

See also

Effect of changes to the type/instance synchronization with control modules (Page 225)

Index

"

A

B

D

E

F

G

M

N

O

P

R

S

T