SIEMENS

SIMATIC

Process Control System PCS 7
CFC for SIMATIC S7

Function Manual

Legal information

Warning notice system

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

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

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

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

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

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

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

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

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 adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the 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.

Table of contents

What's new in CFC? 1

CFC V7.1

Version V7.1 contains the following enhancements or changes compared to the previous version V7.0 SP1:

- **Query of the F password**

  The F password is now no longer queried at the first compilation and download. Instead, they are queried when one of the following applications are run:

  - Opening the object properties of an F-block, F-parameter or F-runtime group
  - Moving F-blocks/groups in the runtime editor
  - Copying, pasting, deleting or interconnecting F-blocks
  - Copying, pasting, deleting or opening of F-charts
  - Updating F-block types

- **Forcing values by connecting the input using "Force"**

  Using the "Force" function, connected block inputs can be forced to take the force value instead of the value of the connection. Forcing can be activated and deactivated in the object properties of the I/O.

  You can find detailed information under Forcing block I/Os (Page 210)

- **AS-wide interconnections**

  AS-wide interconnections can be set up between the projects of a multiproject or for multiple AS of a project. The configuration is performed as it is for a chart-wide interconnection.

  You can find detailed information under Setting up AS-wide interconnections (Page 125)

- **Write protection for charts**

  The write protection can be activated in the object properties of the chart or the object properties of the chart folder / hierarchy folder for all charts contained within.

- **Versioning**

  The properties dialog box opens with the "Version" tab when the modified chart closes if versioning has been activated for the project.

CFC V7.0 SP1

Compared to the previous version V7.0, version V7.01 SP1 contains the following enhancements or modifications:

* **Multiuser engineering**

    Specific improvements were implemented in CFC for multiuser engineering in order to safeguard system integrity and data consistency if several users access the same data. Multiuser engineering is also possible outside of the CFC. For detailed information, refer to the section Configuring on the network system (Page 35).

* **Information function for the number of process objects (PO) used in the project**

    You can call a function in the SIMATIC Manager that identifies all configured process objects which have been registered in Automation License Manager. You can find detailed information under PCS 7 License information (Page 23)

* **Checking and registering process objects (PO)**

    In order to license process objects for an OS and AS, a PO is created in the ES database on each CPU to which a program that contains process objects was downloaded. The identifiers and number of process objects used are registered in this PO information. All existing, registered and missing process objects are check and registered accordingly at each download. For additional information, refer to Counting and registering PO Licenses (Page 24)

Introduction

<div align="right" style="font-size:2em">2</div>

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 are not valid for target systems that are different than 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 tasks required to create a project and download it to the AS in the section Getting started (Page 13).

Getting started

<div align="right">

3

</div>

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 CFC charts**

   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

<div align="right">

4

</div>

4.1 CFC in the STEP 7 environment

STEP 7 components

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

- SIMATIC Manager
  Used as the central database and to coordinate the tools and objects of SIMATIC Manager 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 in 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.
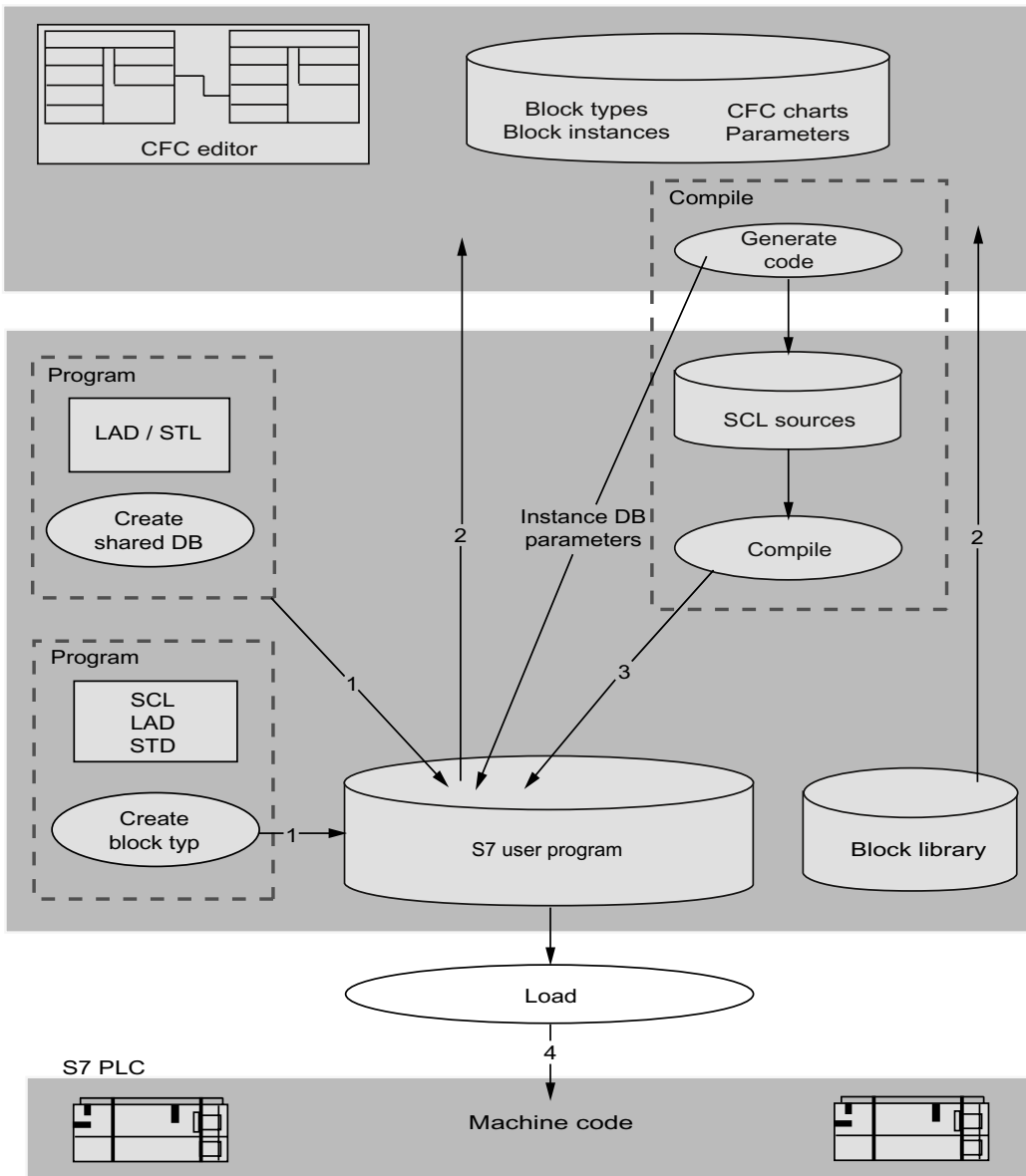
Data flow for configuring an S7-CPU

1. (Optional) Create the block types with STEP 7 tools (for example, with SCL) and save them to the S7 user program.
   (Optional) Global data blocks are created with the LAD/STL editor and stored in the S7 user program.

2. In the CFC editor, create charts in which blocks (taken from the S7 user program or block libraries) are inserted, configured and interconnected.

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.

4.2   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.

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, can only adopt the values 0 or 1.

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

You can find information about other data types in

Data types for S7 (Page 144)

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). Those OBs/tasks are used to execute specific program sections at certain times and in certain situations. 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.

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.

They are displayed in light gray and without visible block I/Os until they are repositioned to a free sheet location. Their interconnections and entries in the sheet bar are also hidden, but nevertheless exist.

4.3   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 edit box.

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.

What you should know about CFC

<div style="text-align: right; font-size: 3em;">5</div>

5.1 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, operating 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.

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.

Show process object statistics

Procedure:

1. Selects the multiproject or the project.

2. Select the PCS 7 license information shortcut menu command.

3. In the left window of the dialog box, select the "Process objects (AS RT PO)" license.

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.

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

For information pertaining to scenarios when counting POs, refer to Counting and logging PO licenses (Page 24)

5.2 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 that was created 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 less POs than the one previously downloaded then the number of the 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.

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.

- **Shutting down 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. Thereafter 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 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 informed if 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.

Additional information pertaining to process objects (PO) is available in the section:

PCS 7 license information (Page 23)

5.3   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.

5.4 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 additional information in the LAD/STL/FBD online help, for example, help on STL, statement list, floating-point functions, master recipe operations.

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

5.5   Migration to the new CFC version

5.5.1   Migration to other CFC versions

Additional processing

You can still use data created with CFC V3.x, CFC V4.x, CFC V5.x, or CFC V6.x as follows in CFC V7.x:

- You can read and display CFC V3.x/V4.x data with CFC V7.x.

- If you edit those data, the contents of each chart folder are converted after you have confirmed the first write access. This ensures compatibility.

You can still use data created with CFC V1.0 or CFC V1.1 as follows in CFC V7.x:

You can read and display CFC V1.x data with CFC V7.x. Direct editing of CFC V1.x data is not possible here. If you want to edit V1.x data, proceed as follows:

1. Select the project in the SIMATIC Manager .

2. Select the menu command **File > Save as...**.

3. In the dialog box "Save File As", select the option "Rearrange before saving (slow)" and save the project under a different name.

    All data are automatically converted to V7.x format.

General notes on compatibility

- You cannot "reconvert" V7.x data to previous versions.

- You cannot use previous CFC versions to edit data created in newer versions.

5.5.2   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 about 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. Update 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 old S7 program name to the new S7 program.

13. Cut the new S7 program and
paste it to the old program of the CPU.
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

5.5.3 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 at which you want to append the runtime group of the next chart 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).

5.5.4 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

<div align="right">6</div>

6.1 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 resides 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 35)"

- PCs not connected to a network
  The project is split into several segments for distributed editing 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 additional information in:

  - Concept and use of installation pointers (Page 165)

  - Textual interconnections in branched and merged project data (Page 138)

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 to the source 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.

6.2 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. 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

Mutual interference may occur when several engineers work on the same chart. **This work procedure is therefore not recommended**.

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 login or logout 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.

Starting and operating the CFC editor

7

7.1 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.

7.2 Operator controls

7.2.1 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)

7.2.2 Catalog of blocks, charts 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 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

- Library catalog

- If the project contains unplaced blocks, the catalog of unplaced blocks is also shown. This catalog is not available as by default.

Where do I find the required object?

If you want to search for a block, proceed as described in Searching for objects in the catalog (Page 45).

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 appear as:

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.

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

You can also insert an empty text box 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 selecting it using 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 top chart is the chart displayed in the SIMATIC Manager . The nested charts contains 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**.

Library catalog

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

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 ![icon] CONTROL)

  and

- In alphabetical order in the "All blocks" folder ![icon]

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

You can enter a block search string in the catalog of blocks or libraries and start the search by clicking ![icon].

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 can find detailed information on this in the section Searching for objects in the catalog (Page 45).

If you want to close all folders, click ![icon].

7.2.3 Searching for objects in the catalog

Searching in the catalog

You can specify a text in the input box of the catalog and search for it by clicking 🔍. 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.

If you click 🔍, you can search for other objects with these letters. The search is completed when it returns to its starting point.

7.2.4 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.

7.2.5 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.
A click the button triggers the function. Disabled buttons are grayed out.

7.2.6   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.

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

The right section 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), 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.

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

7.2.7 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

7.3 Keyboard and mouse operation

7.3.1 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>
- The underscored 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 underscored letter in the name of the required menu command (for example, <N> for the menu command "New").
  A submenu 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>

7.3.2   Shortcut keys in the CFC chart

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

Shortcut key	Function
Chart window	
<Tab>	A **single block** is selected: Moves the cursor to the next block (determined by the internal sequence, 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>	**I/O is selected:** Moves the cursor to the next higher I/O (if it exists, otherwise the block will be selected).
<Arrow down>	**A block selected:** Selects the first input (or first output, if only outputs exist). I/O is selected: Moves the cursor to the next lower I/O (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 or no objects are selected.
<Esc>	Cancels all existing selections in the chart (also multiple selections and the flashing function).

Shortcut key	Function
Interconnections:	
<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
<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:	
<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).
<Return>	In the "Charts" window: Opens the selected chart.
<Return>	In the "Libraries" window: Imports the selected block and places the block instance in the active chart.
<Return>	In the "Unplaced Blocks" window: Places the selected block in the active chart.
<Return>	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 highlights the first object at this level. 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).

Shortcut key	Function
Runtime editor window:	
<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.
<Arrow right> <Arrow left>	Expands the hierarchy of the selected object and highlights the first object at this level. 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).

7.3.3 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

7.3.4   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 underscored character X
Underscored 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

International/German keyboard (Page 58)

7.3.5 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 edit box (left to right and top to bottom)
<Shift+Tab>	Move one edit box back
<Alt+X>	Move to edit box that contains the underscored 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

7.3.6   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

For more information, refer to the section:

International/German keyboard (Page 58)

7.3.7   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 return you to the CFC editor

7.3.8 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 SIMATIC Manager's "Open" dialog box appears.

- 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.

7.3.9 Mouse operation

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; all other objects deselected	Open "Properties – Block" or "Properties – Chart" dialog box	Selection block/ nested chart	Move block/ nested chart
Block body	Select block; all other objects deselected	-	Selection block	Move block
Runtime properties box in the block header	Select block; all other objects deselected	Open "Run Sequence" window	Selection block	Move block
Input, output	Select input/output	Open "Properties – Input/Output" dialog box (or, with STRUCT, "Select Structure Element" dialog box)	Selection block/ nested chart	Move interconnection (rewire)
Nested chart body	Select object; all other objects deselected	Open nested chart	Selection 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 "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
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 in the section Shortcut menu (Page 47).

7.3.10   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>	<Ins>
<BACKSPACE>	<Back>
<Up> <Down> <Left> <Right>	<Up arrow> <Down arrow> <Left arrow> <Right arrow>

Layout of configuration data

<div style="text-align: right; font-size: 2em;">8</div>

8.1 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 60) each with six sheets (Page 60).

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**.

8.2 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 six sheets (Page 60).

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".

8.3 Sheets

Each chart (Page 59) (chart partition) consists of six sheets. In the CFC editor they are arranged in two columns, each containing three sheets. A sheet consists of a central work area and sheet bars (Page 67) 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 $\boxed{1 \; \blacktriangledown}$ 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".

8.4 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.

8.5 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 is displayed.

8.6 Blocks

Display in a chart

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. Blocks positioned so that they partly or fully overlap other objects or the sheet bar are displayed in their normal size, but without block I/Os and in light gray.

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 by the S7_visible: = true attribute. 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.

Block layout

The blocks are displayed in the chart graphically as "large blocks" and "small blocks".

The graphic layout of a large block features the following:

- Header:
  - Block name, block type and comment
  - Field for runtime properties with highlighting of blocks that do not execute.
  - If an interconnected, hidden I/O exists, a colored triangle is displayed in the top right corner.

- Body:
  - "Inputs" and "outputs".
    The I/Os are displayed as selectable fields that contain only the I/O name (narrow layout) or the I/O name and the data type (wide layout).
  - 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 chart contains chart I/Os.

You can set the width of large 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 that you can then use to create new chart folders. Each user can therefore define an individual logon setting.

The graphic layout of a small block is similar to that of a block header of the large block, the difference being that it includes the I/O.

The small block is displayed without boxes or I/O names for the I/Os 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.

8.7 Texts

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

Drag-and-drop the text box 📄 "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 box 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 box, the text cursor is positioned where you clicked.

8.8 Interconnections

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 additional information in the section Working with textual interconnections (Page 134).

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 chart contains chart I/Os.

8.9 Layout of interconnections

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.

Interconnection colors:

The CFC editor displays interconnections, block/chart I/O points to inputs and outputs as well as the sheet bar entries in color. The default color varies according to the data type or group of data types. This feature enhances the chart layout and makes it easier to configure connections.

You can customize the colors of the various objects using the menu command **Options > Customize > Colors...**.

8.10 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)
  Line 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 I/O (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 CPUs.

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).

8.11 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.

8.12 Views

8.12.1 Views in CFC

Views

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

● The overview (Page 71) is useful for copying/moving objects and inserting large blocks.

● Since some details (I/O names, for example) cannot be displayed in this view, certain functions are only available in the sheet view (Page 72).

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

● Click ⊞ (overview) or ▣ (sheet view) on the toolbar

or

● Double-click a free location in the chart

or

● Select the sheet number in the list box 1 ▾ to open the required sheet view or the asterisk (*) to change to the overview.

8.12.2 Overview

Overview

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

- Click in the toolbar.

  or

- Double-click a free point in the chart in the sheet view.

  or

- Select the asterisk (*) in the list box .

8.12.3   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

- Click ▨ in the toolbar.

  or

- Select the required sheet number in the list box of the toolbar

  or

- Double-click a free area of the chart in the overview

  or

- Select the sheet number in the list box [1 ▼].

Creating runtime structures

<div style="text-align: right; font-size: 3em;">9</div>

9.1 Working with charts

9.1.1 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 section Automatic naming (Page 21).

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 84).

Deleting a chart

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

● Select the chart and press <Del>.

9.1.2 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 icon ![icon].

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.

9.1.3   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 list boxes on the toolbar

Toolbar Icons

You can use the Overview [icon] and Sheet View [icon] icons on the toolbar to toggle the 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 [1 ▼].

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:

- **Insertion Position**
  This function starts the runtime editor and displays the block selected from the chart in the run sequence.
  This menu command will be disabled if more than one/no block has been selected.

- **Next Insertion 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 when an overflow page is displayed.

- **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 / connection 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 tacking

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 connection lines associated with a chart partition flash on the following pages:

- Overview page

- All six sheets

- All overflow pages

9.1.4 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 134).

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 reasons for this are described in the section What you should know about copying (Page 111) 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.

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.

9.2 Editing charts

9.2.1 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.

- **Created on**

  Date of creation

- **Last modification on**

  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-protection" 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. You cannot set a number that is smaller than the last saved version.

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.

9.2.2   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 empty. If the chart partition contains any objects, you are prompted to confirm deletion.

9.3 Creating nested charts

9.3.1 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 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 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 on 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 additional information about nested charts in the following sections:

Copying/deleting nested charts (Page 86)

How to replace nested charts (Page 87)

How to create a chart with chart I/Os (Page 88)

9.3.2   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**.

9.3.3 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.

9.3.4   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 the chart I/O names, attributes and default values.
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 on the 🔲 icon on the toolbar
   or
   select the menu command **View > Chart I/Os**.
   The dialog box for editing chart I/Os opens and "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 list 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 note the following:

– 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, i.e., 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 on the ▣ icon on the toolbar
   or
   select the menu command **View > Chart I/Os**.
   The dialog box for editing chart I/Os opens and "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 the start of the line of the chart I/Os in the right window 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 in the following sections:

Compile chart as block type (Page 187)

Interconnection rules for chart I/Os (Page 140)

System attributes for block interfaces

9.4 Handling block types

9.4.1 Import of block types

Import to CFC

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

There are two ways of importing blocks:

- Import in the background: By installing them in the chart from the catalog
- Explicit import: By installing them with the menu command **Options > Block Types…**

Explicit importing 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 "New block type version (Page 93)"<.

Importing using the SIMATIC Manager

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

1. Open the relevant 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. Using different block types (symbolic name and/or function) with identical object names is not permitted, for example, both "FB61", 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. A dialog box appears where you can rename the block, in other words, change the number. 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 an identical 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 additional 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: No. of FBs = 512. The highest possible block number is thus FB 511.

9.4.2 New block type version

Introduction

If you want to use the modified blocks for previously used block types following a block type change, you must import the new block types to the chart folder. The block instances used in CFC are then adapted to the modified block types.
There are two ways of importing:

● Import in the background, achieved by inserting a modified block type from the S7 program or the block library into the chart

● Import using the 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 will be 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 repeated OS compilation because 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, thus requiring OS compilation Example: After you have edited a message text

Changes to the structure 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 on the CPU, 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 chart. Importing is not necessary."

- A list of identical block types

You can find additional information about block type changes in

Effects of type changes on block instances (Page 106)

9.4.3   Inserting block types

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 ding additional information on searching for a block in the section "Searching for blocks in the catalog" (Page 45)

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 "Compare attributes" to open the "Update 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 into 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 appears as follows:

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 appear as follows:

Tip: Inserting blocks from the block families is the fastest method of positioning blocks in the chart, since these are already known in 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 MW0 will be modified to allow blocks called in OB 100 to recognize the startup. MW0 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 on this under "Handling blocks with identical numbers" in the section Import of block types (Page 91).

9.4.4 How to update block/SFC types in the multiproject

Introduction

In the multiproject, you can 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 type synchronization is performed centrally.

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 SIMATC Manager or the blocks or SFCs you want to update in them.

2. Select the menu command Options > Charts > Update Block Types....
   Page 1(3) "Select the S7 program to be checked" of the dialog box opens. This lists all the S7 programs in the multiproject and they are already displayed as selected.

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 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.

   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 are modified** (no additional consequences!).

   – **The OS has to be recompiled**
     All associated instances will be modified. 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 the entire program in STOP operating mode**
     After compiling, you can no longer download changes in the RUN operating mode. All associated instances are modified.

   Note: When updating SFC types on this page 2(3) the button "Display Differences" is shown. 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 indicates which objects have already been updated.
    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 synchronization

    –   The log path

    –   The library

    –   The selected objects

    –   The selected S7 programs

    –   The updated objects

    –   End of type synchronization

7. 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 in

How to update block/SFC types in single projects (Page 99)

9.4.5 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 type synchronization 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 **Options > Charts > Update Block Types....** menu command.
   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 97) (starting at point 3).

9.4.6   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
Lists all block types contained in the chart folder which are not used in the charts, in other words, no instance blocks exist.

2. "Clean up blocks in block folder" dialog box
Lists all block types that exist offline in the block folder that are not used in the CFC (the block types do not exist in the chart folder).
This could also involve 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".

9.5 Editing blocks

9.5.1 Editing blocks

The following sections contain information about editing blocks:

Inserting blocks (Page 102)

Specifying object properties (Page 108)

How to copy blocks (Page 113)

How to move blocks (Page 114)

Deleting blocks (Page 116)

How to align blocks (Page 117)

How to set I/O parameters (Page 118)

What you should know about global block type changes (Page 105)

9.5.2 Inserting blocks

9.5.2.1 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 generate as many block instances as required 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 additional information about inserting blocks in:

Catalog of blocks, charts and libraries (Page 41)

Searching for blocks in the catalog (Page 45)

Settings for inserting blocks (Page 103)

Positioning large blocks (Page 104)

Inserting unplaced blocks (Page 104)

9.5.2.2   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 about this in the section:

Runtime properties of blocks (Page 162)

9.5.2.3 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.

9.5.2.4 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 [ ]. 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.

9.5.3   Global block type change

9.5.3.1   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 on this in the following sections:

Effects of type changes on block instances (Page 106)

Tolerant type import after data type extension (Page 107)

9.5.3.2 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.

9.5.3.3 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.

9.5.4 Specifying object properties

9.5.4.1 Specifying object properties

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

Entering block names (Page 108)

Entering comments (Page 109)

Assigning block icons to specific instances (Page 109)

Specifying the number of I/Os (Page 110)

9.5.4.2 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. Illegal 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 mark)
- 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 box.

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.

9.5.4.3 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 box.

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.

9.5.4.4 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 box, using a maximum of 16 characters.

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

9.5.4.5 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 box 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.

9.5.5 Copying blocks

9.5.5.1 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.

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 in the following sections:

9.5.5.2 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 (Page 70) or open another chart window that displays the destination sheet.

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 icon for **Edit > Copy**.
   The selected blocks are highlighted.

3. Activate the chart to which you want to copy the blocks and

   click the **Edit > Paste** icon.

   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. You can now 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 information on the insertion position in the run sequence in the section What you should know about copying blocks (Page 111).

9.5.6 Moving blocks

9.5.6.1 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 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.

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 block(s) to the new position in the other chart.

Cut and paste

1. Select the blocks to be moved and
   click the **Edit > Cut** ✂ icon.
   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** 📋 icon.
   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.

9.5.7 Deleting blocks

9.5.7.1 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 on this in the following sections:

Deleting block types (Page 100)

Runtime properties of blocks (Page 162)

9.5.8 Aligning blocks

9.5.8.1 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 will be 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 (displayed in light gray).

9.5.9 Editing I/Os

9.5.9.1 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 then select the menu command **Edit > Object Properties...**.
   This opens the "Properties – Block" or "Properties – Chart" dialog box.

2. Select the I/Os tab.
   This tab contains a list of all the inputs and outputs. The list is shown in table format.

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. 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 on the I/O
or
select the desired I/O and select the menu command **Edit > Object properties...**.
The "Properties - Input/Output" dialog box opens.

The dialog elements such as input boxes 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)

● Active (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 more information on this in the section Shared declarations (Page 149).

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".

9.5.9.2 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 toolbar icon
   or
   select the **Edit > Invert Input** 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 note the following:

• 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.

9.5.9.3 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", system attributes of the type "S7_string_0" through "S7_string_25" that may exist at the same I/O are not effective.

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 151).

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.

9.6   SFC in the CFC

9.6.1   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".

9.6.2   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).

9.7 Creating and working with interconnections

9.7.1 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. For more information, see Setting up AS-wide interconnections (Page 125)

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 note the following:

- 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.

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 layout of interconnections in the section

Layout of interconnections (Page 65)

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

How to interconnect to shared addresses (Page 128)

Working with textual interconnections (Page 134)

How to create and delete interconnections with runtime groups (Page 139)

Creating interconnections to chart I/Os (Page 140)

Interconnection rules for chart I/Os (Page 140)

Interconnection rules for data types A, DT, S, and ST (Page 146)

Interconnecting with SFC Charts (Page 133)

9.7.2   Setting up AS-wide interconnections

General information on AS-wide interconnections

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 so 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 100ms. 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 data types STRING, ANY, POINTER and DATE_AND_TIME are not supported.

● A maximum of 200 AS-wide interconnections are permitted per CPU.

● An AS can communicate with a maximum of 15 partner ASs.

● 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.

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. 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 chart). This then generates a warning in the code generator and is not loaded into the AS.

You can find information on eliminating of one-sided AS-wide interconnections under Synchronizing AS-wide interconnections (Page 127)

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 must not be changed.

To view the NetPro IDs, proceed as follows:

- In the CFC, click on the ![symbol] symbol to select the chart reference data.
- In the "View" menu, select the menu item "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.

9.7.3 Synchronizing AS-wide interconnections

Synchronizing 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.

9.7.4 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 toolbar icon).

   The address selection list appears at the I/O, consisting of:

   – The input box (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 box 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 box

   or

   – Select the symbol required.

3. Select a symbol or double-click the line of the selected symbol.

4. Press <Return> in the edit box
   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!

NOTICE
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.

Chart update

Select the menu command **View > Update** or press <F5> to open the dialog for updating symbols, addresses and block types in the CFC charts of the active chart folder, based on the symbol table.

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

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 will be generated and the default value of the block type will be set for the interconnection.

    – Output: The interconnection is ignored in the generated code.

- When interconnecting to a symbol, you can also rename the symbol in the symbol table. CFC recognizes the modifications during a chart update.

- 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.

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 box. 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 additional information on interconnections to shared addresses in:

Possible interconnections between symbols and data types (Page 132)

Examples of symbolic addressing (Page 132)

Examples of absolute addressing (Page 132)

9.7.5 Examples of symbolic addressing

Symbolic addresses

The following examples illustrate symbolic addressing:

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 I5.1. To avoid the absolute address with the same name being addressed instead of the symbol, the symbol is enclosed in quotation marks.

9.7.6 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)

9.7.7 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

9.7.8 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.

9.7.9   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), after it was copied or moved 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 closes 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 reference source is deleted.
  The real interconnection is deleted in this case.

- When the internal interconnections to chart I/Os and their interconnection source 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 note the following:

* 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 into a real interconnection, that is, into a normal block interconnection. The small box no longer displays the identifier (yellow triangle).
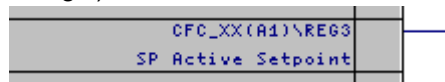
Representation in the sheet bar for AS-wide interconnections

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.
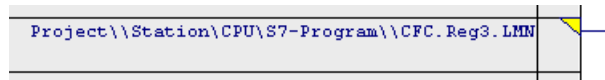
This is converted into a real interconnection by closing the textual interconnection. The yellow triangle in the small field is replaced by a green triangle.
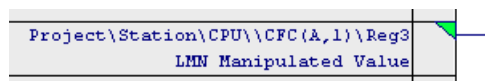
With AS-wide interconnections, care must be given to the syntax of the path in the large field. It is important to distinguish whether or not the chart is integrated in the plant hierarchy.

* The syntax without plant hierarchy

    – Project\\Station\CPU\Program\\Chart\Block.IO

* The syntax with plant hierarchy:

    – Project\\Hierarchy Path\\Chart\Block.IO

Modifying a textual interconnection

You can modify a textual interconnection as follows:

- Rewire it to another input of the same data type.
  You can find more information on this in the section How to modify (rewire)
  interconnections (Page 142).

- Edit the text entry 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 input 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 additional information on this in the section How to interconnect to
    shared addresses (Page 128).

  – Via an interconnection to a chart I/O
    Select the input or text from the sheet bar 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 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 notes

Please note the following:

- 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.

Additional information

You can find additional information on textual interconnections in the section

Textual interconnections in branched and merged project data (Page 138)

9.7.10   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 note the following:

- 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 for additional editing. 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).

9.7.11 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 note the following:

- 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 and then select the
   Insert > Interconnection to Runtime Group... menu command.
   The "Insert Interconnection to Runtime Group" dialog box opens. 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 is 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.

9.7.12 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 additional information about chart I/Os in:

Creating a chart with chart I/Os (Page 88)

Interconnection rules for chart I/Os (Page 140)

9.7.13 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 pointers).
- 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.

9.7.14   Signal tacking

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 (Page 141).

9.7.15   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 appears. 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

Signal tacking (Page 141)

9.7.16   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.

9.7.17   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 note the following:

- 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.

9.7.18   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 respond by clicking "No", 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.

9.7.19   Data types and structures

9.7.19.1   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 lists 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.

ANY, A

ARRAY (not permitted in CFC)

BOOL, BO

BYTE, BY

CHAR, C

COUNTER, CR

DATE, D

BLOCK DB, DB

DINT, DI

DATE AND TIME, DT

DWORD, DW

BLOCK FB, FB

BLOCK FC, FC

INT, I

POINTER, P
REAL, R
STRING, S
STRING[N], SN
STRUCT, ST
TIME OF DAY, T
S5TIME, T5
TIME, TI
TIMER, TR
WORD, W

9.7.19.2 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, FC - IN (with STRUCT)
Target/source of the interconnection (**address**):	Any: CFC block I/O, Address range, DB element, Entire DBs	DB element, CFC block I/O (except for FC - OUT)
Target/source of the interconnection (**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.

9.7.19.3 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 structures

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.

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 chart.

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 chart 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.

9.8 Shared Declarations

9.8.1 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 on this in the following sections:

Configuring enumerations (Page 151)

Configuring units (Page 152)

Configuring equipment properties (Page 153)

Updating shared declarations in the multiproject (Page 155)

9.8.2 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" ... "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. The enumeration with the name "Operating State" is intended to automatically hide messages but does not contain any values.

Create a new enumeration by selecting the "Shared declarations" folder and 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 will find more information in the section Checking the plausibility of shared declarations (Page 157).

9.8.3   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

9.8.4   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 boxes 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:" field, select the equipment property type from the drop-down list.
The instance will now be 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.

9.8.5 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 wish 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 note the following:

- 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...**.

9.8.6 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".

9.9 Configurations for OS and MIS/MES

9.9.1 Operating 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 for OS <-> AS communication, in other words, for operator control and monitoring. You transfer those data to the OS during OS compilation.

Message blocks

After you have inserted a message block into the CFC chart, a message will be generated automatically. Default block message structures are assigned default attributes and message texts. In this way, the CPU sends default event messages 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, call 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 deselecting the "OCM possible" option in the "Object properties" section of the block.

In a WinCC picture template, you can create multiple block icons for each block type. 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 box of the "Object properties" dialog you can define the block instance symbol to be displayed in WinCC. The input box is active if the "Create block icon" check box is selected.

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 consists of the following components:

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

9.9.2 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 box, 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.

9.9.3 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

9.10 Defining runtime properties

9.10.1 Runtime Properties

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 determine the behavior of the target system with respect to response times, dead times, or the stability of time-dependent structures, such as control 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 will find information on installation pointers in the section Concept and use of installation pointers (Page 165). 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.

For more information on attributes, refer to the section:

Task attributes (Page 177)

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.

Additional information

You can find additional information about runtime properties in:

Displaying runtime properties (Page 163)

Modifying the run sequence and the installation pointer (Page 167)

Concept and use of installation pointers (Page 165)

9.10.2 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 180)
   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 insertion position) opens in the run sequence window. The relevant block is highlighted.

2. Select the menu command **Edit > Run Sequence...**
   or
   click on 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 (activated using the menu command **View > Run Sequence** or by clicking the ⬛ icon) displays the full run sequence of the current CPU including comments and runtime attributes. 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.

9.10.3 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 S7 program contains a global chart installation pointer as well as a 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 at the end of existing charts.

- Blocks are inserted in the chart 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 chart created next is to be inserted

- The position at which the runtime group of the chart created next is to be inserted, or after which block at the task level. 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 chart 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 chart (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 of the block after which the new block is to be inserted in the run sequence. The block position is always linked uniquely 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".

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 inserted in nested charts are always 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 32)

9.10.4 Modifying the run sequence and the installation pointer

Starting the runtime editor

Select **Edit > Run Sequence...** or click the toolbar icon [ ] to open another 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 insert instance 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 chart to the required position in the run sequence.

Note

Please note the following:

- 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 Insert 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 Insert 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.

For additional information, refer to the section

9.10.5   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 note the following:

- 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.

NOTICE

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.

9.10.6 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 radio 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.

9.10.7   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.

Three types of runtime groups are used in the run sequence:

- The standard runtime groups resulting when a chart is created or created manually.

- Runtime groups created by the driver generator (@OBxx_@..).

- Runtime groups created by the system in OB 100 (@@OB100@@..).

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. Each runtime group contains only small number of blocks (maximum 10). This avoids aborts during SCL compilation due to the 64K limit.

  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 in the section Runtime attributes (Page 177).

How are runtime groups created?

Runtime groups can be created in two ways:

- **Automatically**

  CFC: After you have created a new CFC chart, 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.

  OB100: If blocks with runtime relevance 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.

  Driver generator: When creating module drivers, runtime groups are created with the identifier "@..." and these can then only be processed by the driver generator.

- **Manually**

  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 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 (you will find information on this in the section Optimizing the run sequence (Page 169))

   – 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.

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.

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.

9.10.8   Task attributes

Attributes

A runtime group is assigned the following attributes:

- Enable (Page 178)
- Reduction ratio (Page 179)
- Phase offset (Page 179)

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.

CAUTION
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.

9.10.8.1 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.

9.10.8.2 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 processed as often as specified by "n", offset in each case by "m" units of the task cycle. "m" is an integer, where $0 <= m <= (n-1)$.

Default: 0, in other words, no phase offset.

Additional information

You can find additional information in the following sections:

Reduction ratio (Page 179)

Example of reduction ratio and phase offset (Page 179)

9.10.8.3 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, in other words, 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 in the following sections:

Phase offset (Page 178)

Example of reduction ratio and phase offset (Page 179)

9.10.8.4 Example of reduction ratio and phase offset

Run-time group with scan rate = 8, phase offset = 3
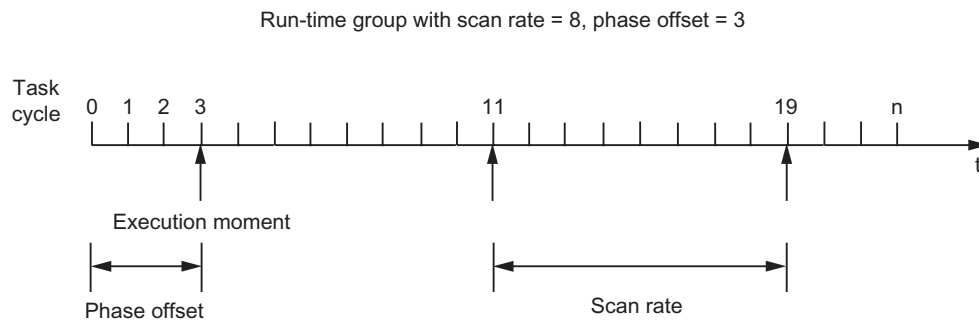
9.10.9 Display of block execution

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 [OB35 1/1] (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 [! OB35 1/1] (red box with a yellow 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 [? OB35 1/2] (yellow field with a black question mark)
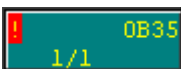
  The following blocks are identified in this way:

  – 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 statuses:

- Block is executed [OB35 1/1] (no identification),

  if the EN of the block and the EN of the group is set to "1".

- Block is not executed [! OB35 1/1] (red box with a yellow exclamation point),

  if the EN of the block or the EN of the group is set to "0".

Compiling

<div align="right">

10

</div>

10.1 Essentials for CFC chart compilation

General

If you want to control the assignment of resources on the target system via the Settings for compilation/download (Page 182) dialog box, it is important to understand the relationships explained below.

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:

- In CFC you can generate SCL source files that exceed the capacity of the SCL compiler. This may occur if too many blocks are installed in a task or runtime group. The FC containing the block calls will thus become too large and the SCL compiler returns the error message "Internal symbol list is full".

  Remedy: Distribute the blocks across several runtime groups. Smaller FCs are thus generated during compilation, since the blocks of one runtime group are called in their own FC.

- 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.

10.2 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:

  - For local data

  - The number of instance DBs of S7 communication

  - The load memory/work memory This is only checked when the download is carried out.

  - The inserted blocks per runtime group or OB

- 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.

10.3 Compile charts as program

Compiling

If objects have been modified in or added to a chart folder, the SIMATIC Manager displays the closed chart folder with an additional icon ![icon] at the lower left corner. This means that the program still needs to be compiled.

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...**.

To compile, select the menu command **Chart > Compile > Charts as Program...** or click ![icon] on 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 file 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 enabled by default, in other words prior to each compile the driver generator will also be started. In special cases, when the hardware is still incomplete, for example, you can disable the check box so that the function "Generate Module Drivers" 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 note the following:

- The warning events written to the log did not abort compilation, but may 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 in the following sections:

Essentials for CFC chart compilation (Page 181)

Settings for compilation/download (Page 182)

Compile and download objects (Page 200)

10.4   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.

NOTICE

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

You can find additional information in:

System attributes of blocks

System attributes for block interfaces

Assigning system attributes

Creating block types in CFC (Page 233)

Downloading 11

11.1 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.

After you have compiled but not yet downloaded the program, the SIMATIC Manager displays an additional icon at the lower left corner of the closed chart folder .

The program is downloaded to the CPU assigned to the active chart.

General notes

Please note the following:

- 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 Object and Download**.
    You can find information on this in the section Compile and download objects (Page 200).

  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 in the section The change log (Page 217).

- 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 on the icon [icon] 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.

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, 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 its blocks 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 full 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, in other words, provided that a download of changes remains possible.

The following applies when performing a full 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 full download has been canceled, it is no longer possible to download changes until a full download has been completed.
  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 stamp is different or data blocks have been added.

● 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 note the following:

● It is always possible to download changes if the following conditions are met:

  – When a full download has been successfully completed 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 195).

● An "asynchronous compression of the CPU" is implicitly performed following the download of changes. 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 entire 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 194)

- 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 195)

NOTICE
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.

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 in the section System support for avoiding causes of CPU STOP (Page 197).

Displaying modifications 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.

VXM opens 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.

11.2   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

What you should know about downloading changes (Page 195)

11.3   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.

NOTICE
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.
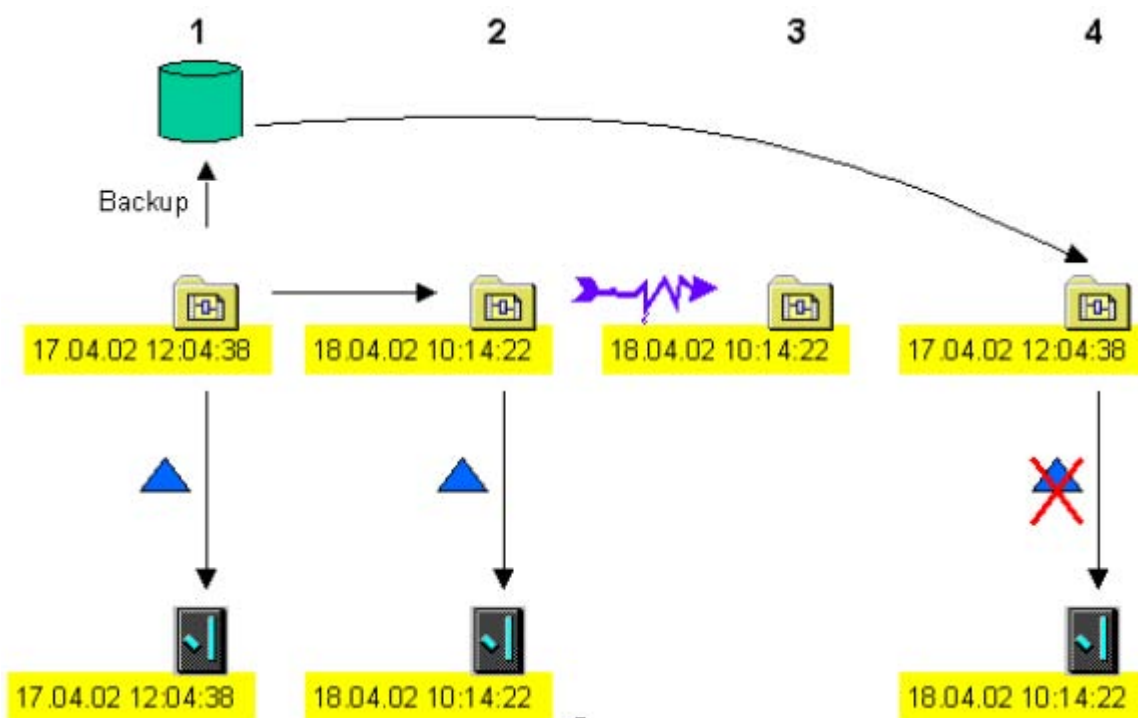
Legend:
Blue triangle→ Download changes

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 it has been modified 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:
You are modifying a program that is now inconsistent, for example, after data was lost due to a power loss.

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).

11.4 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):
  NDmax = NDmax of the OB + NDmax of OB121 + NDmax of OB122

● 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 was 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 resources of the online CPU receiving the download. The system status list is read for corresponding information. A warning message is output in the log if the warning limit is exceeded, however, the download is not canceled. The download is canceled and an error message is output in the log if the absolute limit is exceeded.

● 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 a corresponding message is once again output to the dialog box. 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. A warning message is output in the log if the warning limit is exceeded, however, the download is not canceled. The download will be canceled and an error message is output in the log if the absolute limit is exceeded.

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.
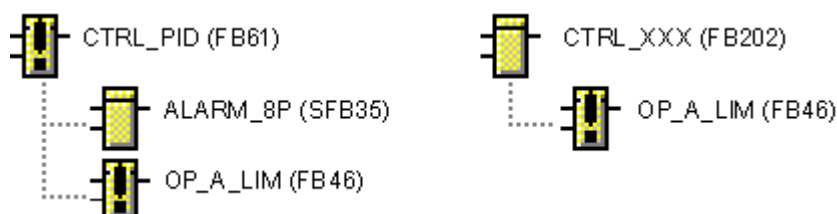
  From this point onwards, the user must intervene:
  Read the chart reference data by selecting the **Options > Chart Reference Data...** menu command.
  Search for the called block by selecting the **Edit > Find** menu command in the "Block Call Hierarchy" view until all calling blocks were found. You 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.

11.5 Compile and download objects

The SIMATIC Manager function "Compile and Download Objects" 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.

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 additional information on this function in the STEP 7 Basic Help.

11.6   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 boxes.

● 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 mode. However, you must ensure in this case that the operating mode remains unchanged until the download is completed.

Testing user programs

<div style="text-align: right; font-size: 3em;">12</div>

12.1 How to work in test mode

Requirement

Prior to test and startup, the user program created in the CFC editor must first have been 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 **Debug > 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 chart.

Editing modes

The CFC editor provides two editing modes:

- Create 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 "Debug" 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 section The change log and ES log (Page 217).

Activating/deactivating test mode

Activate test mode as follows:

Select **Debug > Test Mode**

or click ⬚ on the toolbar.

You can now enable the test menu functions; most of the edit mode functions will be disabled.

Note

If the user program is modified after the code has been compiled a corresponding message is output if you want to change to test mode.

When you **close** the test mode, the test functions will be disabled and the editor functions will be re-enabled.

Exit test mode by selecting the **Debug > Test Mode** menu command or by clicking the toolbar icon ⬚ again.

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 206).
Before the modified value is adopted on the CPU, its validity is 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.

12.2 Monitoring and assigning parameters to block/chart I/Os

12.2.1 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 the debug mode, the "Watch On" function [icon] will be enabled for CFC charts in "laboratory mode". 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 [icon] will be enabled. 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 monitoring. To do this, select the block/chart followed by the menu command "Watch Off".

NOTICE
If watch is disabled and you select an I/O via [icon] in order to register it for testing, monitoring will be enabled for **this** and for **all previously registered** I/Os of this block.

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 note the following:

- 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. 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 OCM-capable, for example, if they are interconnected, the values are shown on 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

Additional information

You can find additional information in the following sections:

Adding/removing I/Os in the watch list (Page 208)

Enable/disable monitoring of I/Os (Page 209)

Configuring I/O (Page 209)

The dynamic display window (Page 213)

The trend display window (Page 215)

12.2.2 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. To do this, select the I/O followed by the menu command **Debug > Inputs/Outputs > Add to Watch List** or click 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 **Debug > Inputs/Outputs > Remove from Watch List**, or by clicking on the toolbar. This is only possible in test mode and if "watch" is enabled.

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".

12.2.3 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

- Via the menu command **Debug > Watch On** or by clicking ⌖ on 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 watch function using the menu command **Debug > Watch Off** or by clicking ⌖. 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)

12.2.4 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 206).
Before the modified value is adopted on the CPU, its validity is checked.

Note

Note that the parameters modified in test mode are always written to the CFC database.

12.3 Forcing

12.3.1 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.

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 box 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.

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 chart** 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".

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.

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.

After forcing was disabled, the block, the system chart, and the runtime group are removed again from the program in the next compiler session.

Data types

The following data types can be forced:

BOOL, BYTE, INT, DINT, REAL, STRUCT, WORD, DWORD

Chart inputs/outputs cannot be forced.

12.4 Dynamic display

12.4.1 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 list 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 I/O required followed by the menu command **Test > I/Os > 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.

12.5 Trend display

12.5.1 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 edit box. 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

<div align="right">

13

</div>

13.1 Change log and ES log

The change log

The change log is opened when an action is performed that needs to be logged if

- The SIMATIC Logon Service is installed
- Access protection and the change log are enabled 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 keeps a record of the user, the changes the user makes, the reasons for the change, the time of the change, and the CPU involved.

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...**.

The ES log

In addition to the information of the change log, the actions performed when downloading and current time stamps can be logged in CFC/SFC if

- the ES log is enabled for the current chart folder.

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 dialog box **"Properties Chart Folder", "ES Log" tab**, select the "ES log active" check box.
   The ES log is then 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, reset the "ES log active" check box (by removing the check mark).

Note

Please note the following:

- 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 has been activated, the "Compile and Download Objects" function of the SIMATIC Manager is interrupted prior to the download to each individual CPU by the ES log opening.

Logging in the ES

Open the 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 consecutively 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 action "Test mode on" is performed, all subsequent actions resulting in a change in values in the CPU are logged. Specifically, these are:

● 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).

Reading back charts

<div style="text-align: right; font-size: 2em;">14</div>

14.1 How to save parameters (read back)

Read Back

You can save (read back) the parameters of all CFC charts 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 full 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 **Chart > Read Back...** menu command.

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 "Offline program"
   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 data store.

   You can select the following in the dialog box in both cases:

   - If you want to read back **all** parameters of the block inputs

   - If you only want to read back the parameters assigned the system attribute for **operator control and monitoring**
   (S7_m_c := "true") or with special **identifier** for the readback
   (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.

● 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 interconnected to the 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

Signal processing

<div style="text-align: right; font-size: 3em;">15</div>

15.1 How to generate module drivers

Generating module drivers automatically

PCS 7 provides a signal processing function that automatically generates required module drivers and interconnects them and sets their parameters accordingly after you have configured the hardware using HW Config and the CFC system functions. 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 in the section Signal processing with driver concept up to V5.2.

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 228), 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 note the following:

- 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.

Additional information

You can find additional information in the following sections:

Brief description of blocks used (Page 226)

How the "Generate Module Drivers" function works (Page 229)

Session model for "Generate Module Drivers" (Page 231)

Devices supported by the driver generator (Page 225)

15.2 Devices supported by the driver generator

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_DIAG1 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.

15.3 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 three 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 preparation of ET 200S motor starter modules.

  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 (see figure Interconnection model, module drivers (Page 228)), 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 a diagnosis of the connection, 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 in:

15.4 Interconnection model, module drivers
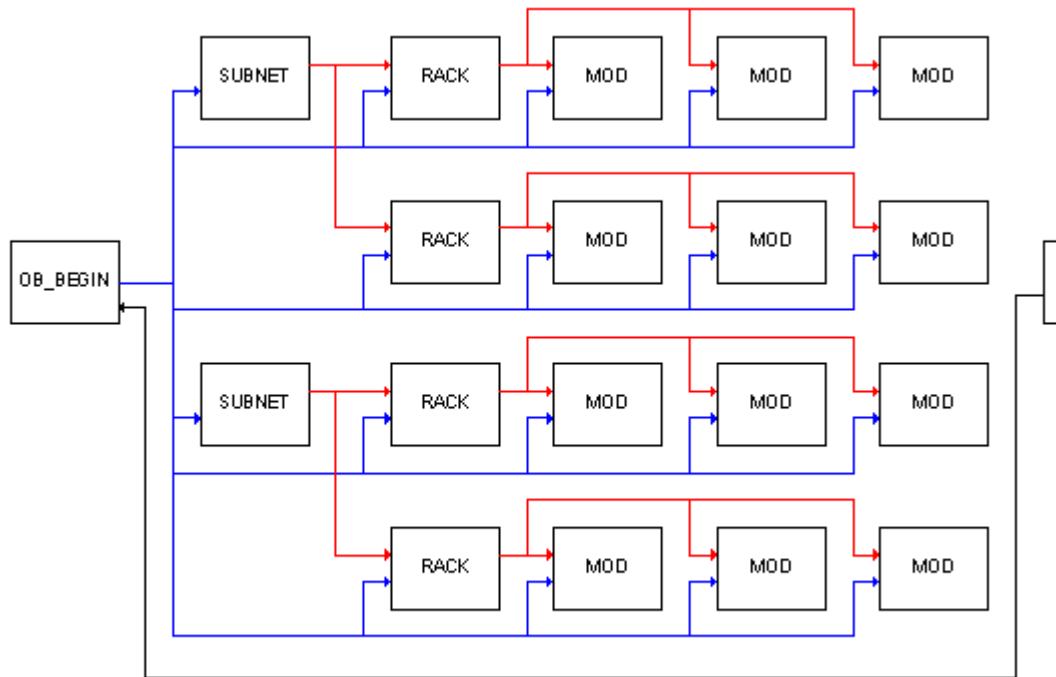
15.5   How the "Generate Module Drivers" function works

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 note the following:

*   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 will be 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 will be assigned the reduction ratio 16. User-specific changes will be 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, i.e., 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.

- 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 in the following sections:

Generate module drivers (Page 223)

Session models for "Generate Module Drivers" (Page 231)

Interconnection model, module drivers (Page 228)

Devices supported by the driver generator (Page 225)

15.6 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 will 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.

Creating CFC block types

<div style="text-align: right; font-size: 3em;">16</div>

16.1 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: No. 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 manipulated.
  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.

Documenting Programs

<div style="text-align: right; font-size: 3em;">17</div>

17.1 Printing charts, dynamic display, I/Os

17.1.1 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.

17.1.2   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 properties.	$60
$$A$$	Author	Name as entered in the chart properties	$55
$$DC$$	Created on	Date as entered in the chart properties	$56
$$DM$$	Last modification on	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+…7+. The number of the last page is appended a "-" character, for example, 127-.

17.1.3   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.

17.1.4   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.

17.2 Printing chart reference data

17.2.1 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 additional information in the chart reference data online help under What you should know about chart reference data.

17.3 Logs

17.3.1 Displaying log files

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.

The different dialog box tabs are described below.

The "Compile" tab

This tab lists the messages that occurred during compilation, including those returned by the compiler, for example, if only the S7 program exists in the project and there is not yet a station: "W: The program is not assigned to a defined CPU. The system will not verify that the CPU is ready to receive the program."

CFC V5.2 and higher evaluates the block reference lists during compilation. If the program uses blocks that contain no reference data (blocks compiled in versions earlier than STEP 7 V5.0 + SP3 or SCL V5.0 + SP3), the check cannot be completed and an error message is output,
for example, "Local data requirements cannot be determined".

Note

CFC version < V5.2 does not output this warning message, in other words, the user had to check local data/memory requirements. If projects containing old blocks without reference lists are edited with a CFC version >= V5.2, the user has to carry out this check, as was previously the case. This prevents excessive local data or memory requirements from stopping the CPU when changes are being downloaded.

Blocks without reference lists are indicated in the CFC chart reference data. To display the chart reference data, select the menu command **Options > Chart Reference Data** . Select the block call hierarchy in the chart reference data window with the menu command **View > Block Call Hierarchy** .
Blocks that do not have a reference list are indicated in this hierarchy as follows:
 No call information

Remedy: Use the latest block library (CFC Library V5.1/PCS 7 Library V5.1 or higher) or compile blocks you have created yourself using a compiler >= V5.0 + SP3.

The "Check Consistency" tab

This tab lists the messages returned by the consistency check. Example: "The immediate successor to the control block SFC_CTRL in the run sequence of OB 31 is not the assigned SFC chart.."

The "Download" tab

This tab lists the messages that occurred during a download, for example, if no errors occurred during the download: "0 error(s) and 0 warning(s) found".

For warnings due to missing block reference lists, see the description of the "Compile" tab above.

The "Block Types" tab

This tab displays the block type log that shows the results of a block type verification (when importing). The log might contain, for example "E: <FB46 OP_A_LIM> cannot be deleted. There are still blocks of this type in the charts".

The "Make Textual Interconnections" tab

This tab displays the textual interconnections closed by this action and error messages returned due to textual interconnections that could not be closed for one reason or another.

The "Module Driver" tab

This tab lists all messages generated when the module drivers were created (SIMATIC Manager > **Options > Charts > Generate Module Drivers...**).

Example: "A: No signal-processing blocks found. Module drivers were not generated."

"ES Log" tab

All actions relevant for validation are logged in this tab when the SIMATIC Logon Service is installed.

Each action is logged consecutively in chronological order. The most recent action appears in the first line.

When the action "Download entire program" is performed, the change 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.

For the action "Start test mode", all subsequent actions resulting in a change (of value) in the CPU are logged. In the CFC this specifically includes new parameters for the I/Os and the activation or deactivation of runtime groups.

The "Read Back" tab

This tab logs all error and warning events, as well as actions taken during read back. This can be, for example, modified I/O values that were read back. These are parameter values to be transferred to the chart folder after they were modified in the CPU.

The "Step Processing" tab

After verifying the SFC runtime behavior with the menu command **Options > Check Step Processing**, this tab lists the steps that use the same address in the terminating action of one step and in the initializing action or processing action of the next step. The log shows how many SFC charts were checked and how many accesses were found. It also shows the SFC charts with unchanged runtime behavior.

The "Convert Format" tab

After converting charts of older versions to V5.1 or higher, the charts that no longer have the same properties are listed here. These include, for example, SFC charts that had the attributes "reduction ratio" and "phase offset" in the old version. The information includes the insertion location (task) and the values for the reduction ratio and phase offset for every chart affected.
Example: "SFC1: Task OB 35 reduction ratio 4 phase offset 2"

The "Details" button

Details are available for specific log messages.

Select the required entry and click "Details". This button is disabled if the message does not contain any information.

Index

M

N

O

P

Time stamp, 201
    Comparing, 201
Tolerant type import after data type extension, 107
Toolbar, 45
Tooltips, 69
Trend display, 215
Type import, 197

U

Unit, 118, 152
    Configuring, 152
    Entering, 118
Unplaced blocks, 104
    Inserting, 104
Update, 97, 99, 184
    Block types in single projects, 99
    Block type in the multiproject, 97
    Sampling time, 184
    SFC types in multiproject, 97
    SFC types in single projects, 99
Updating shared declarations in the multiproject, 155
USB drive, 27
User program, 189
    Download to target system, 189

V

Value identifiers, 121
    Setting, 121
Variable number of I/Os, 110
Version, 81
Views in CFC, 70

W

Watch cycle, 203
Watching I/Os, 209
    Switching on/off, 209
What's new?
    CFC V7.0 SP1, 10
    CFC V7.1, 9
Windows, 40, 213, 215
    Dynamic display, 213
    Trend display, 215
Working, 134, 203
    In test mode, 203
    With textual interconnections, 134
Working window, 40