

# SIEMENS

## SIMATIC

### CFC for S7 Continuous Function Chart

#### Manual

Preface, Contents	
Essentials of CFC	<b>1</b>
First Steps	<b>2</b>
Working with the CFC Editor	<b>3</b>
Test and Commissioning	<b>4</b>
Documentation	<b>5</b>
<b>Appendix</b>	
Technical Specifications	<b>A</b>
Abbreviations	<b>B</b>
Glossary, Index	

## Safety Guidelines

This manual contains notices intended to ensure personal safety, as well as to protect the products and connected equipment against damage. These notices are highlighted by the symbols shown below and graded according to severity by the following texts:



---

### Danger

indicates that death, severe personal injury or substantial property damage will result if proper precautions are not taken.

---



---

### Warning

indicates that death, severe personal injury or substantial property damage can result if proper precautions are not taken.

---



---

### Caution

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

---

---

### Caution

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

---

---

### Notice

draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

---

## Qualified Personnel

Only **qualified personnel** should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

## Correct Usage

Note the following:



---

### Warning

This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

---

## Trademarks

SIMATIC®, SIMATIC HMI® and SIMATIC NET® are registered trademarks of SIEMENS AG.

Third parties using for their own purposes any other names in this document which refer to trademarks might infringe upon the rights of the trademark owners.

### Copyright © Siemens AG 2003 All rights reserved

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

### Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

# Preface

## Purpose of the Manual

This manual "CFC for S7" provides you with the information you require to use the CFC configuration tool in conjunction with CPUs in SIMATIC S7 programmable controllers (PLCs).

For a complete description of CFC for other systems, you also require the supplementary CFC descriptions for the particular system (for example, "CFC for M7", "CFC for SIMADYN D")

## How Sections for Specific Systems are Indicated

If sections, paragraphs or even individual sentences in this S7 manual relate solely to S7 users, this is indicated by **[S7]**. This means that the information is relevant only to S7 or is different in other systems. In this case, if you use a different PLC, you will find the information you require in the manual for your specific system. If the **[S7]** label is in a title, the entire section applies only to S7; if the label is at the start of a paragraph, the paragraph is solely relevant to S7. In lists, the **[S7]** label applies only to the particular list.

## Audience

This manual is intended for personnel involved in configuring, commissioning, and service.

## Validity

This manual is valid for CFC software version 6.0 and higher.

## Standard

The CFC software is based on the international standard DIN EN 61131-3 (IEC 1131-3) for programming languages for programmable logic controllers.

## Conventions

References to other documentation are indicated by numbers in slashes /.../. Based on the number, you can check the full title of the documentation in the References at the end of the manual.

## Further Support

If you have any technical questions, please get in touch with your Siemens representative or agent responsible.

<http://www.siemens.com/automation/partner>

## Training Centers

Siemens offers a number of training courses to familiarize you with the SIMATIC S7 automation system. Please contact your regional training center or our central training center in D 90327 Nuremberg, Germany for details:

Telephone: +49 (911) 895-3200.

Internet: <http://www.sitrain.com>

## A&D Technical Support

Worldwide, available 24 hours a day:



<p><b>Worldwide (Nuernberg)</b> <b>Technical Support</b></p> <p>24 hours a day, 365 days a year          Phone: +49 (0) 180 5050-222          Fax: +49 (0) 180 5050-223          E-Mail: adsupport@siemens.com          GMT: +1:00</p>		
<p><b>Europe / Africa (Nuernberg)</b> <b>Authorization</b></p> <p>Local time: Mon.-Fri. 8:00 to 17:00          Phone: +49 (0) 180 5050-222          Fax: +49 (0) 180 5050-223          E-Mail: adsupport@siemens.com          GMT: +1:00</p>	<p><b>United States (Johnson City)</b> <b>Technical Support and Authorization</b></p> <p>Local time: Mon.-Fri. 8:00 to 17:00          Phone: +1 (0) 423 262 2522          Fax: +1 (0) 423 262 2289          E-Mail: simatic.hotline@sea.siemens.com          GMT: -5:00</p>	<p><b>Asia / Australia (Beijing)</b> <b>Technical Support and Authorization</b></p> <p>Local time: Mon.-Fri. 8:00 to 17:00          Phone: +86 10 64 75 75 75          Fax: +86 10 64 74 74 74          E-Mail: adsupport.asia@siemens.com          GMT: +8:00</p>
<p>The languages of the SIMATIC Hotlines and the authorization hotline are generally German and English.</p>		

## **Service & Support on the Internet**

In addition to our documentation, we offer our Know-how online on the internet at:

<http://www.siemens.com/automation/service&support>

where you will find the following:

- The newsletter, which constantly provides you with up-to-date information on your products.
- The right documents via our Search function in Service & Support.
- A forum, where users and experts from all over the world exchange their experiences.
- Your local representative for Automation & Drives via our representatives database.
- Information on field service, repairs, spare parts and more under "Services".

# Contents

<b>1</b>	<b>Essentials of CFC</b>	<b>1-1</b>
1.1	General.....	1-1
1.2	[S7] CFC in the STEP 7 Environment.....	1-2
1.3	The CFC Chart.....	1-3
1.4	Blocks in CFC.....	1-6
1.5	The Catalog.....	1-9
1.6	[S7] Operator Control and Monitoring.....	1-10
1.7	Steps in Configuration.....	1-11
<b>2</b>	<b>First Steps</b>	<b>2-1</b>
2.1	Creating a Closed-Loop Control with a Simulated Process.....	2-2
2.1.1	Creating the Project.....	2-2
2.1.2	Creating a Chart.....	2-3
2.1.3	Compiling and Downloading the Chart.....	2-6
2.2	Testing the Program.....	2-7
2.3	Making Changes to the Chart.....	2-9
2.3.1	Changing the Run-Time Properties.....	2-9
2.4	Creating Chart I/Os and a Chart-in-Chart.....	2-13
2.4.1	Creating a Chart with Chart I/Os.....	2-13
2.4.2	Inserting a Chart in Another Chart.....	2-16
2.5	Creating a Block Type.....	2-17
2.5.1	Testing the Block.....	2-18
<b>3</b>	<b>Working with the CFC Editor</b>	<b>3-1</b>
3.1	Handling Charts.....	3-2
3.2	Creating a Chart.....	3-4
3.2.1	Adapting Chart Properties.....	3-4
3.2.2	Inserting and Deleting Chart Partitions.....	3-4
3.2.3	Creating a Chart with Chart I/Os.....	3-5
3.2.4	Creating Nested Charts.....	3-7
3.3	Handling Blocks.....	3-9
3.3.1	[S7] Importing Blocks.....	3-9
3.3.2	[S7] Importing a New Version.....	3-10
3.3.3	Effects of Type Changes on Block Instances.....	3-12
3.3.4	Inserting Blocks in the Chart.....	3-14
3.3.5	Copying and Moving Blocks.....	3-16
3.3.6	Deleting Blocks.....	3-16
3.4	Editing Blocks.....	3-17
3.4.1	Setting Object Properties.....	3-17
3.4.2	Changing the Number of I/Os.....	3-18
3.5	Modifying the Properties of Inputs and Outputs.....	3-19
3.5.1	Inverting a Block Input.....	3-19
3.5.2	Value Identifiers.....	3-20

3.6	Interconnections .....	3-21
3.6.1	Interconnections to Shared Addresses .....	3-22
3.6.2	Interconnections to Run-Time Groups .....	3-22
3.6.3	Textual Interconnections .....	3-23
3.6.4	[S7] Interconnecting with SFC Charts (CFC in PCS 7).....	3-26
3.6.5	Handling Interconnections.....	3-27
3.6.6	Structures .....	3-29
3.7	Run-Time Properties of the Blocks .....	3-30
3.7.1	Sequence Model of CFC.....	3-31
3.7.2	Modifying the Run Sequence and the Installation Pointer .....	3-33
3.7.3	Run-Time Groups.....	3-35
3.7.4	Optimization of the Run Sequence .....	3-37
3.8	[S7] Generate Module Drivers.....	3-38
3.9	[S7] Compiling.....	3-42
3.9.1	[S7] Compile Charts as Program .....	3-42
3.9.2	[S7] Compile Chart as Block Type .....	3-45
3.10	Downloading the User Program to the Target System .....	3-46
3.11	[S7] Reading Back Charts.....	3-49
3.12	Several Engineers Working on one Project .....	3-50
3.12.1	Textual Interconnections When Branching and Merging Project Data .....	3-51
<b>4</b>	<b>Test and Commissioning</b> .....	<b>4-1</b>
4.1	General.....	4-1
4.2	Functions Before and During the Test .....	4-2
4.2.1	Comparing the Time Stamp of the CPU Program.....	4-2
4.2.2	Starting and Stopping the CPU Program .....	4-2
4.2.3	Clearing/Resetting a CPU .....	4-3
4.2.4	Set Time and Date .....	4-3
4.2.5	Displaying Module Information.....	4-3
4.3	Working in the Test Mode .....	4-4
4.4	Monitoring and Assigning Parameters to Block I/Os .....	4-6
4.4.1	Block and Chart I/Os in the Chart Window .....	4-7
4.5	The Dynamic Display .....	4-9
4.5.1	I/Os in the Dynamic Display Window .....	4-10
<b>5</b>	<b>Documentation</b> .....	<b>5-1</b>
5.1	Printing a Chart .....	5-1
5.1.1	Footers .....	5-2
5.2	Chart Reference Data .....	5-3
5.2.1	Lists of the Chart Reference Data.....	5-4
5.3	Logs.....	5-4
<b>A</b>	<b>Technical Specifications</b> .....	<b>A-1</b>
A.1	[S7] Technical Specifications .....	A-1
A.2	Field/Name Lengths and Conventions.....	A-2
A.3	[S7] Data types.....	A-3
<b>B</b>	<b>Abbreviations</b> .....	<b>B-1</b>

**Glossary**

**Index**

# 1 Essentials of CFC

## Introduction

This chapter provides you with basic information about CFC, shows how it fits into the STEP 7 software package, describes the block concept, and explains the steps required from creating the project structure to testing the program.

For a description of installation, authorization, and starting up the CFC software, refer to the readme file shipped with the CFC package.

## 1.1 General

### What is CFC?

CFC (Continuous Function Chart) is a graphic editor that can be used in conjunction with the STEP 7 software package. It is used to create the entire software structure of the CPU from ready-made blocks. When working with the editor, you place blocks on function charts, assign parameters to them, and interconnect them.

Interconnecting means, for example, that values are transferred from one output to one or more inputs during communication between the blocks.

### How the Package Works

In the CFC editor, you work with graphic tools: You select ready-made blocks from the pool of blocks, drag them to the chart (that serves as your drawing board) and then interconnect them using the mouse. You do not need to be aware of details such as algorithms or the assignment of machine resources but can concentrate solely on the technological aspects of your configuration.

The run-time properties of the blocks have default settings but these can be adapted individually for each block. Since individual blocks or whole 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.

Once you have created all the functions you require, you can create executable machine code with a simple mouse click, download the code to the PLC, and test it with the CFC test functions.

## 1.2 [S7] CFC in the STEP 7 Environment

### SIMATIC Manager

The SIMATIC Manager is used for all PLCs as the graphic interface to coordinate the tools and objects. The SIMATIC Manager manages tools and data and is used, for example, for creating and modifying a project structure (CPU, CFC charts) and to start the CFC Editor.

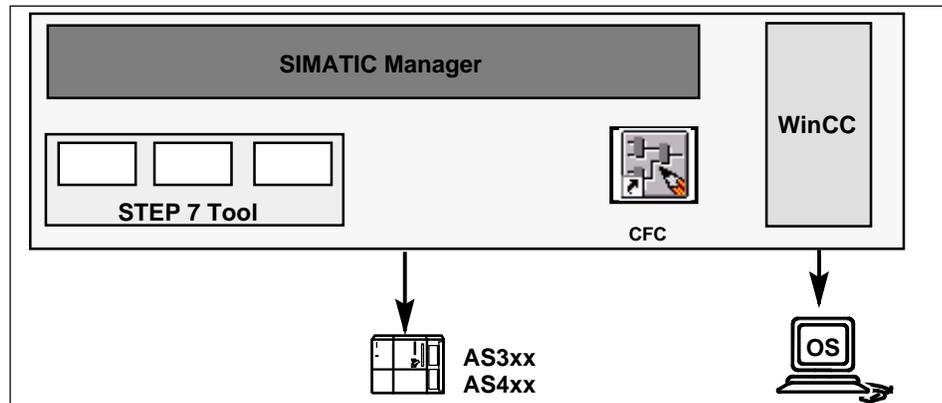


Figure 1-1: CFC in the STEP 7 Environment

### Further Components

Depending on your programmable controller, you can use further components, for example different language packages for creating block types and tools for creating input data for the CFC charts such as I/O data that you can reference from within CFC.

## 1.3 The CFC Chart

To establish the terminology we will be using, the section below describes the CFC chart and its elements.

### Charts and Chart Partitions

The basic working unit of the CFC editor is the chart. Each chart has a name that is unique in the CPU. You can create charts either in the SIMATIC Manager or directly in the CFC editor.

Each chart partition consists of up to 26 chart partitions. A newly created chart consists of a single chart partition, further chart partitions can be added.

### Sheets and Sheet Bars

Each chart consists of six sheets arranged in two columns of three in the CFC editor (see Figure 1-2). A sheet consists of a central working area and sheet bars containing the sheet and interchart references. On the working surface, you can position and interconnect blocks or further charts.

### Overflow Page

If there are so many interconnections to other sheets, so that not all entries can be included because the sheet bar is full, an overflow page is created automatically. The overflow page is an extension of the sheet bars and contains no further objects.

### Nested Charts

A CFC chart can be inserted in another CFC chart (chart-in-chart technique). This allows hierarchical structures to be created. Each inserted chart can be opened just like any other chart, edited, and individually modified.

A chart can be "encapsulated" for further use; in other words, it is given chart I/Os. For each chart, you can decide which block I/Os are available at the chart I/Os.

For further information about creating nested charts, refer to this manual, Chapter 3 or look in the online help.

## Chart Overview and Sheet View

You can change between the chart overview and the sheet view at any time.

The overview is useful for copying and moving blocks/charts and for inserting larger blocks. Since, however, certain details cannot be displayed in this view, for example the names of inputs and outputs, certain functions are only possible in the sheet view.

With the zoom functions, you can change the size of the view in regular steps between the smallest and largest display.

## Display of a Chart

You can customize the display of the various chart elements. With blocks, you can, for example, decide whether the type name or the FB/FC assignment is displayed in the block headers and whether the data type and I/O name or the comment is displayed for I/Os. The width of the blocks and the sheet bars is variable.

For more detailed information, refer to the online help.

## Example of a Chart Overview

Figure 1-2 shows an empty CFC chart (6 sheets) in the chart or overview display.

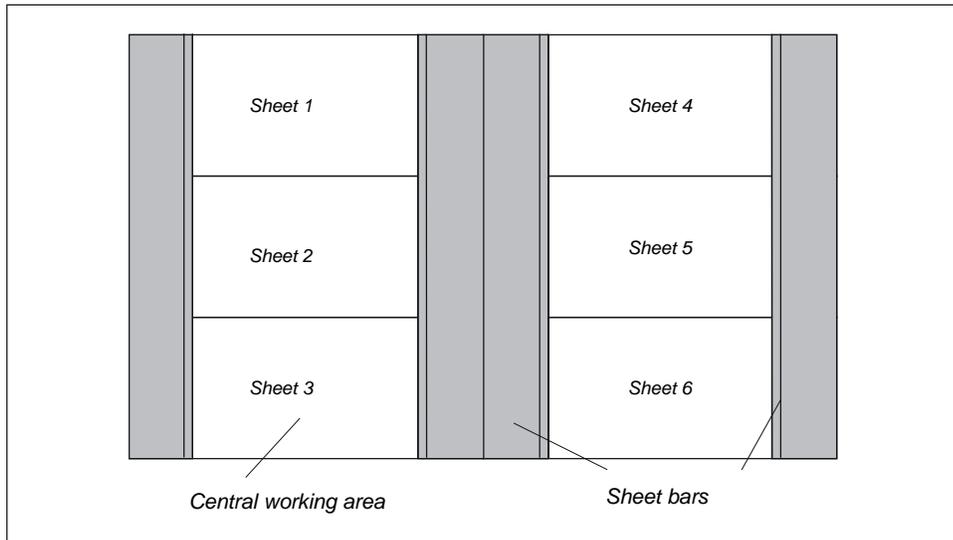


Figure 1-2: CFC Chart in the Overview Display



## 1.4 Blocks in CFC

### Functions in the Form of Blocks

In CFC, you work with ready-made blocks that have a specific function. You place these function blocks in the chart, interconnect them, and assign parameters to them.

### The Block Type

A type definition that specifies the algorithm, the type name, and the data interface (the input and output parameters) exists for each function block.

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

- **CTUD** (COUNT UP and DOWN) for the function of an edge-controlled count up/count down counter.
- **COUNT\_P**, a counter that counts up or down (depending on the setting) at the positive edge of a binary signal.
- **ADD\_R**, a simple function that adds input values and applies the sum to the output.

The type definition also specifies the **data types** of the input and output parameters. These input and output parameters are known as block inputs and block outputs since this is how they appear in the graphic display of the block. The data type of the input or output specifies which values it can adopt, for example,

**BOOL** Boolean type, can only adopt the values 0 or 1,

**STRING** character string type, can contain a string of characters as its value.

There are also other data types, refer to the Appendix, Table A-2.

### The Block Instance

If you now place a block in your CFC chart, by inserting it in your chart, you create a block instance of this block type. Instance in this sense means that it is a usage 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 them, and assign parameters to them without changing the functionality specific to the type.

One advantage of this type instance concept is, for example, that following later central changes to the block type, these changes can be automatically made in all the corresponding block instances.

## [S7] Complex Blocks (multiple instance blocks)

Functions can also be put together using different subfunctions. These subfunctions themselves are blocks and are put together to form a complex block. A controller block can, for example, contain a message block and an operator control block (multiple instance block).

Using CFC, you can create these blocks by interconnecting different blocks (functions) and assigning suitable parameters in a chart. You then compile the chart as a block type (see Section 3.9.2).

## Blocks with a Variable Number of Block Inputs

In CFC, there are blocks whose number of inputs is variable and can be changed in the CFC chart (generic block). A block with a variable number of inputs is, for example, the **AND** block.

## Block Families

Blocks are grouped together according to their functional properties to form block families. When it is created, each block receives a family identifier. The following blocks, for example, make up block families:

- the conversion blocks for adapting various data types  
**CONVERT** (BO\_BY, BY\_DW, W\_DW, ... etc.).
- the multiplexer blocks =  
**MULTIPLX** (MUX8\_R, MUXn\_DI, ... etc.)
- the blocks with math (floating point) functions  
**MATH\_FP** (SQRT, ADD\_R, ... etc.)

The names of the block families are used, among other things as a criterion for sorting blocks in the CFC catalog.

## [S7] Organization Blocks

The interface between the operating system of the CPU and the user program are the tasks known in S7 as organization blocks (OBs). Using these OBs, specific program sections can be executed at certain times and in certain situations. There are OBs for CPU startup (cold restart, hot restart), for process interrupts, for cyclic interrupts (with different time bases) etc.

Organization blocks or tasks are not blocks in the sense understood in CFC; they can neither be inserted nor edited in CFC. After calling the run sequence editor, the blocks placed in the CFC chart are displayed in the OBs in the order in which they are processed..

### Further Distinctions

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

- An **FB** is a block with memory; in other words, the data exist during processing from cycle to cycle and can be accessed. To make this data accessible, a data block (DB) is created for each block instance. In a complex block, the FB has further subsidiary FBs for which only one common DB is created.
- An **FC** is a block without memory; in other words, the values generated by the block are processed immediately. No data block is required here. An FC does not have default values at the outputs.
- A **BOP** (like the FC) is also a block without memory. Basic operations are program components in CFC that are entered as SCL statements during compilation and are used for simple functions such as AND, OR etc.

### Special Case: Overlapping Blocks

Overlapping blocks are blocks that have been inserted or moved to a chart and do not have adequate space. In this case, they overlap other objects completely or partly.

Overlapping blocks are displayed in light gray and without visible block I/Os until they are relocated at a free position on the sheet. Existing interconnections and the entries in the sheet bar of overlapping blocks are also invisible, but nevertheless exist.

## 1.5 The Catalog

### Catalog of Blocks, Charts and Libraries

Blocks and charts that you want to insert in the CFC chart (using drag-and-drop) can be taken from a catalog.

The catalog consists of a window in which the existing block families, librariesLibraries etc. are listed (tree structure). The range displayed depends on the PLC and the installed libraries.

You can switch from window to window using the buttons visible at the lower edge of the window:

	<b>Blocks:</b> Here, you will find the existing (imported) blocks and BOPs sorted alphabetically under "All Blocks" and according to families and the blocks of the current S7 program that you can insert in the chart.
	<b>Libraries:</b> Here you will find the block libraries from which you can insert new blocks into the chart. The libraries known to the SIMATIC Manager are displayed.
	<b>Charts:</b> Here, you will find the CFC charts of the current S7 program that you can insert (copy) or open.
	<b>Unplaced blocks:</b> Here, you will find the blocks of the current program that are no longer displayed in a CFC chart. The CFC chart to which the blocks were assigned is also displayed here. The catalog is displayed only when unplaced blocks exist.
	In the lower part of the catalog, you will find the following buttons:
	<b>Find block or chart:</b> You can specify a block or chart name in the box and search for it with the "Find" button. The folder (for example of the block family) in which the object is located is opened. You only need to type the first few letters. The search stops when an object with these letters is found. During the search, a dialog box with a progress bar is displayed. Here, you can also stop the search if it takes too long.
	<b>Close folders:</b> Below the "Find" button there is a "Close" button. With this button, you can close all the open folders in the catalog.

## 1.6 [S7] Operator Control and Monitoring

During operation of the process, messages are generated on the AS that need to be passed on to the operator control and monitoring system (PCS 7).

With the message configuration in CFC, you can configure event-dependent messages, their texts, and attributes directly in the block.

While configuring the AS, you create data that are required on the OS for communication between the AS and OS; in other words, for operator control and monitoring. You can transfer the data to the OS in the SIMATIC Manager.

### Message Blocks

When you insert a block with message capability into the CFC chart, a message is created automatically. This block has a message structure with default attributes and message texts; in other words, the PLC sends the message when an event occurs without any extra configuration on the part of the user. The signals that form messages can also have associated values that allow dynamic values to be entered in the message texts.

You can edit the attributes (such as message class, message type) and the message texts for the individual block instances with the message configuration functions (Special Object Properties: "Messages"). If the "Messages" button is not activated, the block is not capable of sending messages.

### Operator Control Blocks

All the message blocks for PCS 7 have an attribute for operator control and monitoring (S7\_m\_c). Texts can be selected and edited for blocks with inputs that allow operator input. You can start the dialog with the "Operator Control and Monitoring..." button in the Object Properties of the block. The "operator control and monitoring" attribute of CFC blocks (instances) can also be modified. To modify this property, select or deselect the "Operator control and monitoring" option in the object properties of the block.

### AS-OS Communication

When you have completed configuring the messages, the data required on the OS for communication between the AS and OS is transferred to the OS (OS compilation). This data is transferred to one or more destination operator stations and is used on these stations by graphic objects or faceplates. To make this transfer, the "AS-OS Engineering" software package must be installed.

For more detailed information on transferring data, refer to the online help.

## 1.7 Steps in Configuration

### Order of the Steps

1. Create the project structure
2. Create blocks and import them to CFC (optional)
3. Insert Blocks in the Chart
4. Set parameters for the blocks and interconnect them
5. Adapt Run-Time Properties
6. Compile the CFC charts
7. Download the CFC Program
8. Test the CFC Program

### Create the Project Structure

To work with CFC, you must create a chart folder below the program (folder for the system-specific program) using the SIMATIC Manager.

You create the individual CFC charts in the chart folder either using the SIMATIC Manager or directly in the CFC editor.

**[S7]** When you create the project structure, you are supported by the "New Project" wizard (depending on the setting either the STEP 7 wizard or the PCS 7 assistant). The PCS 7 assistant creates a multiproject in the component view and in the plant view. As default, the PCS 7 assistant also creates a CFC chart.

### Create Blocks (optional)

CFC works with ready-made blocks. These can be blocks from libraries, other programs, or types you created yourself. You will find information on creating blocks in Section 3.9.2, "Compiling as Block Type", and in the manual "PCS 7 Programming Instructions Blocks".

### Import Blocks

The way in which block types are included and, in some cases, imported depends on the PLC.

For more detailed information, refer to Section 3.3.1.

### Insert Blocks in the Chart

Blocks are inserted in the chart by dragging them from the catalog. This 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.

For more detailed information, refer to Section 3.3.4.

## Set Parameters and Interconnect Blocks

You can assign parameters to the inputs and outputs of the blocks or you can interconnect them either with other blocks or with shared addresses. (Shared addresses are connection partners located outside the CFC chart, for example in S7: peripheral I/O signals, memory bits, timers, counters, and shared data blocks.)

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 interconnected partner exists and the interconnection is then made using a menu command.

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

For more detailed information, refer to Section 3.5.

## Adapt the Tun-time Properties

The run-time properties of a block decide how the block is included in the run sequence within the entire structure of the PLC. These properties are decisive for the response of the PLC in terms of reaction times, dead times, or the stability of time-dependent structures, for example closed loops.

When it is inserted, each block is assigned default run-time properties. The block is installed in a task at a position that you yourself can select. You can change the position at which the block is installed and other attributes later if necessary.

For more detailed information, refer to Section 3.7.

## Compile the CFC Chart

During compilation as a program, all the charts of the active CPU are converted to machine code. Different compilers are used depending on the destination PLC; the call is, however, identical. If you compile as a block type, only the individual chart is compiled.

For more detailed information, refer to Section 3.9.

## Download the CFC Program

After compilation, you can download the CFC program to the CPU.

Changes to the program can also be downloaded online without stopping the CPU.

For more detailed information, refer to Section 3.10.

## Test the CFC Program

After compiling and downloading the program you can test it. The range and type of test functions available differs from PLC to PLC. In the test mode, you are connected online to the programmable controller.

For more detailed information, refer to Section 4.3.

## 2 First Steps

### Introduction

This chapter "First Steps" is intended as a primer for newcomers to CFC who want to get to know the package quickly. The example is divided into various tasks and guides you step-by-step from the simplest configuration jobs to the creation of a chart with chart I/Os and blocks in CFC.

---

**Note:**

You will find a ready-made example in the SIMATIC Manager as follows:  
**File > Open...** > "Sample projects" tab > **ZDt04\_01\_CFC** (German).  
(English: **ZEn...**, French: **ZFr...**, Spanish: **ZEs...**, Italian: **ZIt...**)

---

In this example, it is assumed that CFC will be used in the STEP 7 environment. This means that the STEP 7 standard package, SCL, and CFC are installed. The PLC used is either S7-300 or S7-400.

You can create the sample project "CFCEXA\_2" described below with the SIMATIC Manager.

## 2.1 Creating a Closed-Loop Control with a Simulated Process

### 2.1.1 Creating the Project

This section describes the steps involved in creating a project with the menu commands of the SIMATIC Manager. You configure the hardware with HW Config (this can be done later but must be done before you download to the CPU). This example is restricted to the S7 program:

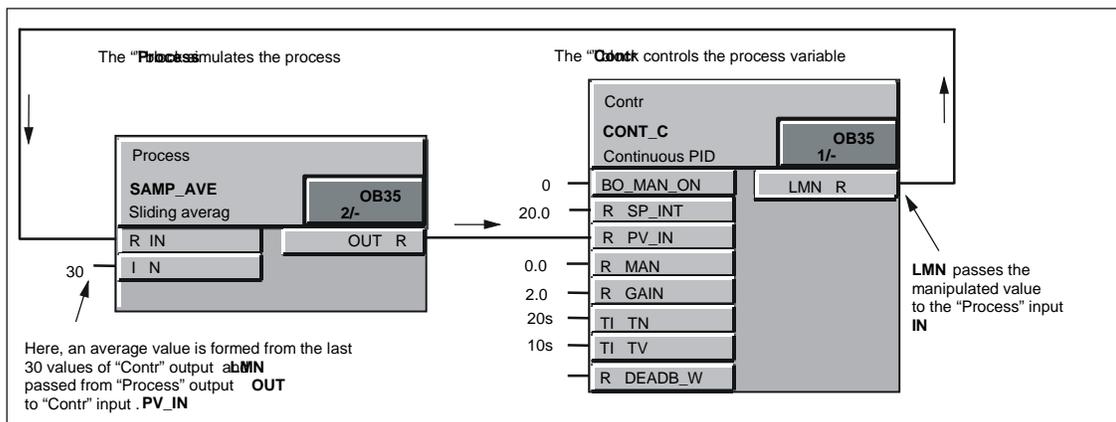
- In the toolbar, select  or **File > New...**  
In the "New Project" dialog box, enter the project name "CFCEXA\_2" and enter it with "OK".
- With the project folder selected, click the menu command **Insert > Program > S7 Program**.  
The S7 program is created in the "Component View" with a source files folder, block folder, and symbol table.
- With the S7 Program folder selected, click the menu command **Insert > S7 Software > Chart Folder**.  
The chart folder is created.
- With the chart folder selected, click the menu command **Insert > S7 Software > CFC**.  
A chart "CFC(1)" is created; Give this the name "Control".
- Double-click the CFC chart to open it.

All the requirements for working with the CFC editor have now been satisfied.

## 2.1.2 Creating a Chart

### Aim

You will now create a controller with process simulation in which the process is simulated by a sliding average value. You will use two blocks for this, SAMP\_AVE and CONT\_C. The SAMP\_AVE block forms the average value from a number of input values and the CONT\_C is a PID controller that controls this variable average value.



### Inserting the Blocks

- Open the catalog  if it is not already open (default).
- In the catalog, click the button  of the libraries. Here you can open the **CFC Library**. This is a collection of block libraries.
- Now open the folder **ELEM\_300**. This is a library with blocks suitable for the S7-3xx CPU. If you are using the S7-4xx CPU, open the folder **ELEM\_400**. You can drag blocks from the list that appears to the chart.
- Click **CONT\_C**, hold down the mouse button and drag the block to the chart. Position it to the top right on sheet 1.
- Then take the block **SAMP\_AVE** and position it on the left beside the CONT\_C block.
- Double-click a free position close to the two blocks to change to the sheet view  (or, in the toolbar click  ).

In the sheet view, you can see the blocks as graphic objects with a header and several I/Os on the body. The I/Os (inputs left, outputs right) are displayed as fields with the I/O name and in the "wide" display the data type. With **Options > Customize > Block/Sheet Bar Width.....**, you can set the block width "Narrow" or "Wide". If "Narrow" is set (default), the blocks are displayed narrower and without the data type being shown.

### Interconnecting the Blocks

Now interconnect the blocks as follows:

- On the SAMP\_AVE block, click the output **OUT** and then click the input **PV\_IN** on the CONT\_C block.
- On the CONT\_C block, click the output **LMN** and then click the input **IN** on the SAMP\_AVE block.  
As an alternative, you can also drag a block output to the input with which you want to connect it using the mouse.

The two blocks are now interconnected.

### Changing the Appearance of the Blocks.

The blocks are displayed in the chart with all their I/Os (inputs and outputs) as dictated by the block type. In our example, however, we do not require all the I/Os and to make the display simpler and clearer we want to make the unnecessary I/Os invisible in the chart. In the same dialog, we will also change the block names.

- Double-click the block header of the CONT\_C block: The "Properties" dialog box is opened for this block. The name ("1") is already selected and you can type in the new name "Contr" immediately.
- Now select the "Inputs/Outputs" tab. Using the horizontal scroll bar, go right until the "Not displayed" column appears.
- Click the first selection cell, hold down the mouse button, and drag the mouse pointer vertically to the end of the column: The entire column is selected. With the mouse pointer in the selected area, click the right mouse button and select the "Set" command in the menu.

All unconnected I/Os are set to "Not displayed". Some I/Os will, however, be needed later in the test mode to input values. We will make these visible again.

- In the "Not displayed" column, click the check boxes of the following:

MAN\_ON  
SP\_INT  
MAN  
GAIN  
TN  
TV  
DEADB\_W.

## Setting Parameters for the I/Os and Selecting Them for Testing

- In the "Inputs/Outputs" tab, go to the column "Watched" and set all the visible I/Os including the interconnected output **LMN**.
- In the "Value" column, enter "20" for **SP\_INT** (this is the default setpoint for the controller). Close the Object Properties by clicking "OK".

You can also set parameters directly for an individual I/O:

- Double-click the block input **MAN\_ON** of the controller.
- In the "Value" box, change the "1" to "0". This disables the "Manual Mode" that would interrupt the control loop.
- Close the dialog box by clicking "OK".

Follow the same procedure with the SAMP\_AVE block (using the Properties dialog of the individual I/Os or in the Properties dialog of the block as described below).

- Double-click the SAMP\_AVE block header. Name this block "Process".
- In the "Inputs/Outputs" tab, set the input **N** in the "Watched" column (if it is not already set).
- In the "Value" column, enter the value "30" for **N**. (This is the number of input values to be used for the average value.)
- Close the dialog box by clicking "OK".

The blocks are now interconnected and have the parameters required for our process simulation.

### 2.1.3 Compiling and Downloading the Chart

The next step is to compile the chart as a program.

- Select the following button in the CFC toolbar  or **Chart > Compile > Charts as Program...**  
In the dialog box that appears, set "Compile: Entire program". Complete the dialog with "OK".  
Compilation is now started and the progress is displayed in a dialog box.  
Confirm the final message with the S7 logs with "Close" (you can ignore the displayed warning).

**Note:**

The next step is only possible if you have configured and connected a CPU of the type S7-3xx or S7-4xx to your PC. The setting of the key switch on the CPU must be: RUN-P.

- To download the program to the CPU, select the button  or **PLC > Download...**  
In the dialog box, select the type of download (this is already set: "Entire program").

Before you download the program, the CPU is set to STOP (after a prompt that you answer with "Yes") and any blocks already on the CPU are deleted. The download is displayed in a further dialog box. After downloading the programs successfully (with no errors), a message is displayed to show that downloading is complete and asking you whether you want to restart the CPU. If you answer "Yes", you can return the CPU to the "RUN" mode.

The CPU changes to the **RUN** mode. The program is loaded and can now be tested.

## 2.2 Testing the Program

In the test mode, you can monitor the values of the block I/Os and change the values of the block inputs. The values registered for testing are shown on a yellow background.

If you change some of the parameters, you can monitor the controller response, for example how the manipulated value approaches the setpoint and settles.

### Changing to the Test Mode

Before you change to the test mode, change the mode from "Process Mode" to "Laboratory Mode" ("Test > Laboratory Mode"). This means that all block I/Os are automatically activated for "Watching".

Note: In the "Process Mode", the default setting is **no** I/O registered for watching. In this test mode, you would have to select the relevant blocks and register them

explicitly for watching (by clicking ).

### Activating the Test Mode:

- Click  or select **Debug > Test Mode**.

### Changing Values Online

You can set a different setpoint for the example, as follows:

- On the controller, double-click the **SP\_INT** input and set a different value (< 100) as the internal setpoint in the dialog box that follows.
- Click "Apply" so that the value is adopted and the dialog box remains open for further changes.

After you have made a few changes and observed the control response, close the dialog box with "OK".

You can, for example, influence the speed of the settling at the block inputs:

**GAIN** (Proportional gain, determines the control gain)

**TN** (Reset time, determines the I-action)

**TV** (Derivative time, determines the D-action)

If you change "GAIN" to a lower value and "TN" to a longer time, the dynamic response of the controller is changed and the control response of the example is more "sluggish".

With the **MAN\_ON** block input, you can interrupt the control loop and switch over to "Manual Mode" (=1). The manipulated value (in other words the value at the output **LMN**) is then set by the value of the **MAN** input.

## **The Result**

In this part of the example, you have got to know the elementary aspects of configuring in CFC. You have created a project with the SIMATIC Manager, created a CFC chart, and inserted blocks from a library. You have interconnected the blocks and set parameters. You have created an executable program and downloaded it to the CPU. In the test mode, you were able to monitor and modify the dynamic response of the control loop.

## 2.3 Making Changes to the Chart

We will now leave the test mode.

- You change to the edit mode by clicking the  button.

### 2.3.1 Changing the Run-Time Properties

#### Introduction

The blocks of a chart have certain run-time properties. These run-time properties determine when and in which order the blocks are executed on the CPU. To structure their execution, the blocks are installed in OBs. In this example, the default installation of the blocks is in OB35 (cyclic interrupt 100 ms) and because they are also involved in a restart, they are installed in OB100 (warm restart).

So that you do not need to worry about the run sequence for every block, CFC installs the blocks one after the other after a particular block. This block is also the "Predecessor for Installation" for blocks installed later. In the CFC status bar (to the bottom right in the window), you can see which block is currently the "Predecessor for Installation".

When you create a CFC chart, a run-time group is created automatically and has the same name as the chart.

You can assign attributes to the run-time group that decide the scan rate of the OB cycle and the phase offset with which the blocks are executed.

#### Changing the Run-Time Properties

You want the blocks to be executed in a different sequence. The test mode is deactivated, you now call the run-time editor with the run sequence.

- Click the button in the toolbar  or select **Edit > Run Sequence....**

A new window is opened displaying all the OBs. Objects have already been installed in OB100 and OB35 as can be seen by the + in a box in front of the OB icon.
- Select OB35 and the run-time group "Ctrl" it contains. The blocks are displayed in the right-hand detailed window.
 

Note: The run-time group was created automatically when you created the chart.
- Keep the mouse pointer on the run-time group and select the **Object Properties...** menu command with the right mouse button. A dialog box is displayed.

- Make the following entries in this dialog box:

Name:	Ctrl	(default retained)
Comment	U8_PV0	
Scan rate	8	
Phase offset	0	(default retained)
Optimizing the Run Sequence	•	(default retained)
Active	•	(default retained)

- Enter your settings with "OK".

With the setting you have made for the scan rate, the blocks are now executed every eighth run; In other words with the basic cycle for OB35 of 100 ms they will be executed every 800 ms.

The phase offset can be used to achieve a better distribution of load on the CPU when you have blocks in several run-time groups. Since this is not relevant in this example, the default "0" remains, in other words, no phase offset..

The "Optimize Run Sequence" option determines whether or not the run-time group is included in an optimization run that must be started explicitly (see Section 3.7.4). With the "active" option, the run-time group is activated or deactivated for processing in the CPU.

## Copying Blocks within the Chart

As a practical exercise, you will now copy the content of sheet 1 to sheet 2 and continue editing there. When you copy interconnected blocks the interconnections are retained.

- Change back from the run sequence to chart editing.

To do this, click any point in the chart window (CFCEXA\_2\S7 Program(1)\...\Control) or click



again and change to the overview by clicking



- In sheet 1, hold down the left mouse button and draw a lasso around the interconnected blocks. The blocks are now selected (blue).
- Remain on the selection with the mouse pointer, hold down the Ctrl key and drag the blocks to sheet 2 (below sheet 1).
- Select the "Contr1" block, copy it and insert it in the same sheet again. The block is called "Contr2".

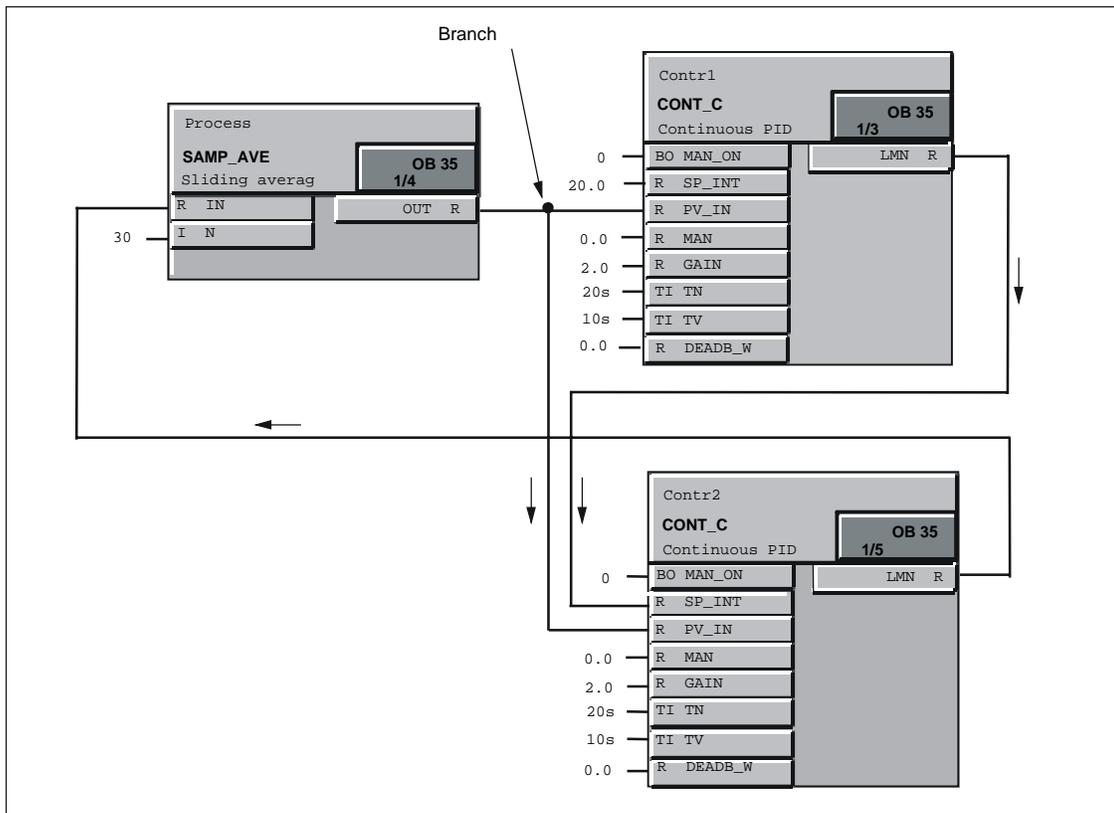
### Changing the Interconnection

- Click the connecting line or the output **LMN** of "Contr1" and press the "Del" key. The connection to input **IN** of "Process1" is deleted.
- Click **LMN** of "Contr1" and then **SP\_INT** of "Contr2".

(Later, you will learn a more elegant method of "rewiring" without deleting and interconnecting again).

- Click **LMN** of "Contr2" and then **IN** of "Process1".
- Click **OUT** of "Process1" and then **PV\_IN** of "Contr2".

With the blocks positioned as shown, the interconnection appears as follows:



## Compiling, Downloading and Testing Changes

The chart must be compiled again and then downloaded to the CPU.

- Select the  button in the toolbar.  
In the dialog box, set the option "Compile: Changes only" and click "OK".  
Compilation is started, confirm the logs message with "Close".
- To download the program, select the  button.  
In the dialog box, set "Download: Changes only" and confirm with "OK".

When you download changes (as opposed to the entire program) the CPU does not need to be set to STOP.

Caution! If you are working with a real project, make sure that you are familiar with the information in "Reasons for STOP when Downloading Changes Online" in the online help.

(**Help > Contents**, "Index" Tab: Type in "Reasons for" and click the "Display" button.)

After downloading, you can return to the test mode and test your modified program.

## The Result

In this part, you have learnt that the blocks of the CFC chart have certain run-time properties on the CPU and that you can modify them. You have also seen that substructures known as run-time groups are used in the run sequence and which attributes you can assign to them.

You have copied blocks within a chart and seen that the interconnections between the blocks are retained. You have modified interconnections and once again created an executable program. You have seen the difference between downloading the entire program and downloading changes only.

## 2.4 Creating Chart I/Os and a Chart-in-Chart

In the following section you will create chart I/Os for a CFC chart and insert this chart in a different CFC chart.

### 2.4.1 Creating a Chart with Chart I/Os

The chart I/Os of a chart can be used to "encapsulate" CFC charts for further use. When you create the chart I/Os, you can specify which block I/Os are relevant for interconnection with other charts or blocks and must be applied to chart I/Os.

#### Preparations

- Create a new chart by clicking  in the toolbar. In the dialog box, enter the object name : "Sim\_reg" and confirm with "OK". The new chart is displayed.
- By clicking on  in the toolbar, the chart "Sim\_reg" and the chart "Control" are displayed one beside the other.
- Set the overview display for both charts by clicking .
- Copy the blocks of sheet 1 of the "Control" chart to sheet 1 of the "Sim\_reg" chart in the same way as you did when copying blocks within a chart.
- Close the "Control" chart and change to the sheet view (sheet 1) of the "Sim\_reg" chart.
- Open the block catalog in the catalog by clicking on the  button and then on the block family **MULTIPLX**.
- Drag the block **SEL\_R** to sheet 1 and give it the name "Switch" (in the Properties dialog).

To include the "Switch" block in our example, you must now "rewire" an existing interconnection; In other words you modify an interconnection without deleting the existing one.

- On the "Contr" block, select the **PV\_IN** input, hold down the mouse button and drag the I/O to "Switch" **IN1**. The output **OUT** of "Process" is now connected to **IN1** of "Switch".

As an alternative, you could also delete the existing connection and create new interconnections.

The output of "Switch" must now be connected to the input for the process variable of "Contr".

- Connect "Switch" **OUT** with "Contr" **PV\_IN**.

The "Switch" now switches depending on the value of the input **K**, the value of the input **IN0** (K=1) or **IN1** (K=0) to output **OUT**.

In a real project, this would allow you to switch over between a process simulation (IN1) and a real process (process value from the process connected to IN0).

## Creating Chart I/Os for the Chart

You now create the chart I/Os for the chart. These are then connected to the selected block I/Os.

- Click the  button in the toolbar or select **View > Chart I/Os**.  
The dialog for editing chart I/Os is opened and "docked" to the upper part of the chart window.
- In the left-hand window, click the Block Icon of the inputs **IN**. The block inputs are displayed in the right-hand window (currently empty).
- In the working field of the chart, select the **MAN\_ON** block I/O on "Contr" and drag the I/O to the right window of the chart I/Os to the "Name" box. The I/O is then entered with all its properties.
- Follow the same procedure with all further non-interconnected inputs (see table).
- Change the name of I/O **K** of the "Switch" block in the chart I/Os by double-clicking in the "Name" box. Enter **SIM** here. Instead of **IN0**, enter **PV** (Process value).
- In the left window of the chart I/Os, click on the block icon of the outputs **OUT**. Select the **LMN** output on the "Contr", hold down the Ctrl key and drag the I/O to the right window of the chart I/Os to the "Name" field.

The chart I/Os then appear as follows:

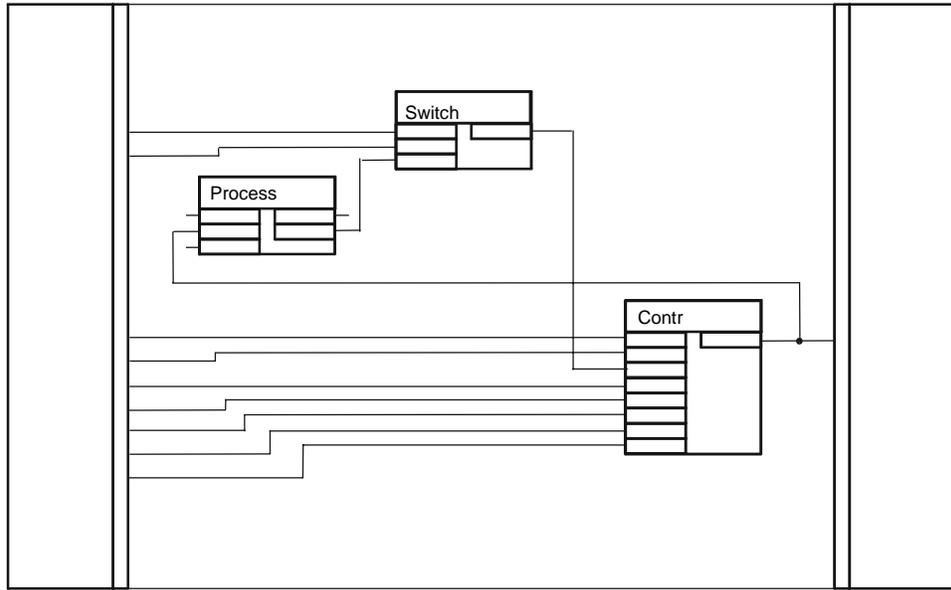
Block	Block input	Data type	Block output	Data type
Contr	MAN_ON	BOOL	LMN	REAL
	SP_INT	REAL		
	MAN	REAL		
	GAIN	REAL		
	TN	TIME		
	TV	TIME		
	DEADB_W	REAL		
Switch	SIM (previously: K)	BOOL		
	PV (previously: IN0)	REAL		

The sheet bar displays the I/O names and comments, I/O type, and data type applied to the chart I/Os.

You have now created all the chart I/Os for the chart.

You can now close the window of the chart I/Os by clicking  again and can "tidy up" the chart to make it clearer to read.

- Move the blocks in the chart so that as few connection lines as possible cross over other lines. One possible arrangement is shown below.



## 2.4.2 Inserting a Chart in Another Chart

The chart "Sim\_reg" created in the previous configuration step with chart I/Os will be inserted in another chart. Create the new chart as follows:

- Click the  button in the toolbar. In the dialog box, enter "Top Chart" in **Object name:** and confirm with "OK". The new chart is displayed.
- Open the "Charts" catalog by clicking the  button. In a tree hierarchy you will see the charts "Top Chart", "Control" and "Sim\_reg". **Note:** If the catalog only contains the message "!" (no hierarchy folder exists) then you have set the option "Display catalog with plant hierarchy" in the **Options > Customize > Display...** dialog box. Since the project was created without the plant hierarchy, this option must be disabled (click the check box: The check mark is removed).
- Select the "Sim\_reg" chart and drag it to the working area of "Top Chart". The original chart is copied.
- Change to the sheet view.

The chart with chart I/Os appears like a block and can be recognized as a chart by its icon:



So that you can see that this is a copy of the previously created chart, open it by selecting it and then selecting the **Open** command using the right mouse button. In the title bar, you will recognize that this is a "nested chart" by the path: ...\\Top Chart\\Sim\_reg.

In the catalog of the charts, a + box is displayed in front of "Top Chart". By clicking the box (or double-clicking the chart icon), you can open up the tree and the hierarchy of the chart becomes visible: The "Sim\_reg" chart is displayed in this branch as an active chart (icon of the open folder).

To return to the top chart, you can select "Open Parent Chart" with the right mouse button or select the path for the "Top Chart" in the "Window" menu.

### The Result

In this part, you have learnt how to edit a chart so that it has chart I/Os that allow it to be interconnected to other block I/Os and to be used as often as required. You have seen how a chart can be inserted like a block with the chart-in-chart technique. You have seen that, in contrast to the block, the inserted chart can be opened and modified.

With the chart-in-chart technique, you can create nested charts and therefore create a structure according to technological aspects with greater clarity.

## 2.5 Creating a Block Type

Normally, the entire chart folder containing the open chart is compiled. This produces a program that can be downloaded to a CPU. You can, however, also compile a single chart and create a block type from it. This is then placed in the block library or in the S7 program so that it can be used again.

### Compiling a Chart as a Block Type

You will now compile the original chart "Sim\_reg" as a block type.

- Close all the charts (**Window > Close All**).

Make sure that you open the **Original chart** "Sim\_reg" that is located at the same hierarchical level as the charts "Top Chart" and "Control".

- Select the "Sim\_reg" chart in the "Charts" catalog and open it with "Open" using the right mouse button.
- Select **Chart > Compile > Chart as Block**.

A dialog appears in which you can enter further information.

- In the "Properties Block Type" box, enter the following:

FB number:	110
Symbolic name:	REG_1
Name (header):	REG_1
Family:	CONTROL
Author:	TEST
Version (Header):	0.1

and confirm with "OK".

The compilation is started and progress is indicated in a dialog. After successful compilation, the "FB110" block is in the block folder and the symbol name "REG\_1" is entered in the symbol table.

## 2.5.1 Testing the Block

The next step is to create a new chart and to insert the block **REG\_1** in it.

- Create a new chart "Test".
- Press the "F5" key (or **View > Update**) so that CFC reads the changes in the symbol table and the block folder.
- Open the S7 program in the catalog of the blocks. Here, you will see the new block type **REG\_1**.
- Insert **REG\_1** into the "Test" chart by dragging it with the mouse and change to the sheet view. You will see the block I/Os as you created them as chart I/Os. The **EN** and **ENO** I/Os are added by the system (so that the block can be activated and deactivated). These I/Os are invisible (default). If you want to display these I/Os as well, you must make them visible in the Object Properties, "Inputs/Outputs" tab.
- Compile the charts as program all together and download the program to the



CPU. Click .

You receive a message that the program has been changed and must first be compiled.

- Answer the question "Do you want to compile now and then download?" with "Yes".

You see a dialog box with the tabs "Compile Charts as Program" and "Download S7".

- Select "Scope: Changes" in both tabs and start with "OK".



- Then change to the test mode to watch and change the I/Os of the block.

With the **SIM** input, you can toggle between internal simulation (= 0) and external process value (of the **PV** input) (= 1).

### Final Comments

In this brief example, you have got to know a few of the possibilities available with CFC. The exercises have illustrated how simply and conveniently you can create a program for an automation task that can then be run on the CPU.

Once you have worked through this example, you will know CFC well enough to start tackling more complex tasks.

The following chapters and the comprehensive online help of CFC will provide you with more information.

# 3 Working with the CFC Editor

## Overview

This chapter describes how to configure an entire software structure for a CPU using the CFC editor.

## Requirements

Using the SIMATIC Manager, you have created a project with a program folder for a specific AS (for example an S7 program for SIMATIC S7) including a chart folder.

---

### Note:

CFC is "upwards compatible", this means that programs modified with other tools and not with the CFC editor lead to inconsistencies.

---

### Restrictions for multi-users in network operation:

Several users can work on one project. This allows configuration, testing, and commissioning of ASs to be performed at different locations or in a PC network (multi-user mode).

If the PCs are connected in a network, remember that one AS must only be edited by one user at any one time.

If several people want to work on a project at different locations, the project can be divided up, edited, and put back together again. You can distribute individual charts of an S7 program in several working projects and then bring them together again after they have been edited separately.

For more information on this topic, refer to Section 3.12, Several Engineers Working on a Project or the CFC online help.

## 3.1 Handling Charts

### Creating a Chart

You normally create a chart with the SIMATIC Manager (Menu command "Insert > S7 Software > CFC"); this is, however, also possible directly in the CFC editor ("Chart > New" menu command). The name must be unique on the CPU (this is checked by the system) and can be a maximum of 22 characters long.

When you create a new chart, a new run-time group is created automatically and installed in the run sequence according to the chart installation pointer. The name of the run-time group matches the chart name.

### Opening a Chart

You can open a chart with the **SIMATIC Manager**. Select a project and the project folder, open the charts folder and double-click the required chart to open it and start the CFC editor at the same time.

In the **CFC Editor**, the "Chart" menu always displays the last four charts that you edited (and closed) as a menu item. If you select one of these names, the corresponding chart is opened or if it is already open it is displayed.

You can open a chart that is not displayed in the "Chart" menu by selecting the "Chart > Open" menu command, selecting the project in the dialog box, selecting the program folder and the "CFC" object type and opening the selected chart by double-clicking it.

## Copying/Moving a Chart

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

When you copy/move charts, the resources (block types, FBs and FCs including symbols and the system attributes and the called blocks of multiple instance blocks), are also copied if they do not already exist in the destination.

Remember that the Copy/Move function has effects on existing interconnections and blocks etc. 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.

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 with the menu command "Options > Block Types..." and make a central block type change with the "New Version" button.

The copied blocks retain the run-time properties of the blocks from the source CPU; in other words, they 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 with the missing tasks is output and the copy function is rejected.

If you copy/move a chart to another chart folder, the interconnections that go to other charts become **textual interconnections**. These open textual interconnections can be closed again (made into real block interconnections) if the chart with the interconnection partners is also copied or moved to this chart folder or if the chart is copied/moved back to the original folder.

If textual interconnections already exist in a chart and the chart is copied/moved, the textual interconnections are taken with the chart.

For more information on textual interconnections, refer to Section 3.6.3.

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

CFC charts can also be copied/moved between different target systems (for example SIMATIC S7 • SIMATIC M7). Once again, the block types used in both target systems must be identical; in other words, they must be compatible since they are not copied.

## Closing/Deleting Charts

Since all the changes in the chart are saved immediately, you can close the chart or exit the editor at any time.

You can delete a CFC chart in the SIMATIC Manager.

## 3.2 Creating a Chart

In its original form (after it has been inserted in the chart folder), a CFC chart consists of a chart partition with 6 sheets without further chart partitions. You can rename such a chart and extend it.

You can add chart I/Os to a chart (see Section 3.2.3) so that, for example, it can be interconnected with other charts or inserted inside another chart where it can be interconnected with blocks. If you use the chart-in-chart technique (inserting charts with chart I/Os in another chart), you create nested charts (see Section 3.2.4).

A chart can also be inserted into another chart without chart I/Os (for example when you want to create the chart I/Os later).

### 3.2.1 Adapting Chart Properties

In the "Properties CFC Chart" dialog box, you can set properties such as the chart name, author, and comment for the active chart.

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

For further information and information about assigning names in PCS 7, refer to the CFC online help.

### 3.2.2 Inserting and Deleting Chart Partitions

You can add further chart partitions to the CFC chart at any time if it is not large enough for your needs. For each chart partition, a tab with which you can display the required chart partition is displayed at the lower edge of the window at the height of the scroll bar.

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. A chart can consist of up to 26 chart partitions; they are identified in alphabetical order (A - Z). The alphabetical identifier of the individual chart partitions can change if you insert further chart partitions.

If, for example, the "CFC1" chart consists of a single chart partition, this is given the letter "A". If you insert a further chart partition before this chart partition, the new "first" chart partition becomes "A" and the previous one now becomes "B".

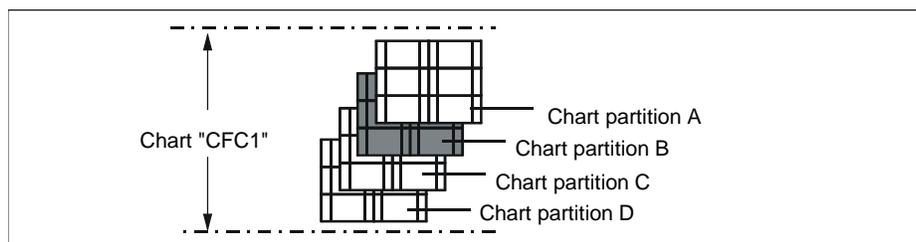


Figure 3-1: Chart with Chart Partitions

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

Example:

Sheet view:  Overview: 

**You can only delete** the active chart partition. If the active chart partition is empty, it is deleted after you select the "Edit > Delete Chart Partition" menu command. If the chart partition contains objects, you will be prompted for confirmation before it is deleted.

### 3.2.3 Creating a Chart with Chart I/Os

You can provide a chart with I/Os to extend your options such as

- Installation in a different chart (nested charts, see Section 3.2.4) and interconnecting with other charts or blocks
- Compiling it as a block type

You assign the selected I/Os of the internal blocks contained in the chart or the nested charts to the chart I/Os.

#### The "Chart Inputs/Outputs" Window

You create the chart I/Os in a separate CFC window.

With the "View > Chart Inputs/Outputs" menu command or the  button in the toolbar, you can open (and close) the "Chart Inputs/Outputs" window.

The procedure for creating chart I/Os is described in detail in the example in Section 2.4.1.

#### System Attributes

Just as in blocks, you can also assign system attributes to the individual chart I/Os of a chart.

A chart with chart I/Os does not have system attributes itself (apart from those of the I/Os). You can assign these attributes when the chart is compiled as a block type (see Section 3.9.2).

**Notes:**

- You can only interconnect unconnected I/Os with the chart I/Os if the I/Os have a compatible data type.
- If an I/O is already interconnected, it is not possible to change the data type.
- Chart outputs cannot be assigned parameters if they are interconnected.  
Remedy: You assign parameters to the output of the block; this value is then entered for the corresponding chart output.
- With inputs, you can assign the chart I/O to several block/chart inputs.

**Assigning I/Os when the Charts are Already Installed**

You can also extend a chart with chart I/Os later by adding further chart I/Os. If this is a nested chart (see also 3.2.4); in other words, a chart that is already installed in another chart, the added I/Os can lead to positioning conflicts. In this case, the nested chart is displayed as an overlapping chart just like an overlapping block; in other words, light gray and without I/Os.

Once the chart is positioned at a free location, the I/Os and interconnections are visible again.

### 3.2.4 Creating Nested Charts

You can insert a CFC chart in another CFC chart. This allows you to create a program structured according to your technological requirements using standardized sections that can be used again and again.

The charts nested in another chart can be opened and modified in the Edit mode and can be run and monitored in the Test mode.

The chart containing further charts is the top chart. In the SIMATIC Manager, you can only see the top chart in the chart folder. The maximum nesting depth for nested charts is 8 (top chart + 7 levels of nested charts).

#### Inserting a Chart in a Chart

You can drag and drop a new nested chart into the current chart from the block catalog (button  New Chart) and continue editing there. If you use the menu command "Insert > New Chart", a free position is looked for automatically in the chart (in the sheet view only in the current sheet) and the new chart is placed at the free position.

To allow charts to be used like blocks in CFC, they are displayed in the chart catalog in a tree structure. From here, 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 its own nested charts (if it contains further nested charts).

You can also move parent charts to the chart by dragging them with the mouse while holding down the Shift key. In this case, they are not copied but moved. In the catalog, the chart is now no longer at its previous location but is shown in the hierarchy of the active chart.

#### Navigating in a Chart

To open a nested chart, select the chart within its parent chart and select the "Open" menu command with the right mouse button or in the "Edit" menu. You can also open the chart by double-clicking a free position in the body (not the header or I/O). By repeating this, you can work down to the lowest chart nested in the hierarchy.

To move up through the hierarchy (as far as the top chart), select the nested chart and then select the "Open Parent Chart" menu command with the right mouse button or in the "Chart" menu.

You can also open a chart in the chart catalog. Select a chart and then select the "Open" menu command with the right mouse button.

#### Copying a Nested Chart to a Different CPU

You can copy a nested chart and paste it in the chart on another CPU. The position at which the blocks are inserted is decided by the "Predecessor for Installation" of the destination chart. Run-time groups are not copied.

## Copying and Deleting

If you want to delete a nested chart 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), you can first copy it into the chart folder. With the chart open (active), select the "Chart > Copy to Chart Folder" menu command. The chart is copied and inserted in the chart folder with a new name. You can now delete the chart within a chart as usual.

## Interconnecting

If the nested chart has I/Os, you can interconnect it in the chart with other nested charts, with blocks, or with shared addresses.

How to assign block I/Os to the chart I/Os is described in Section 2.4.1. You can use the same procedure when you want to connect the chart I/Os of nested charts to chart I/Os of charts higher up in the hierarchy.

You can also make this interconnection if you select a block/chart I/O and then select "Connection to Chart I/O..." with the right mouse button or in the "Insert" menu. The "Chart Inputs/Outputs" window does not need to be open. A dialog box is displayed with the list of all available I/Os; in other words if you have selected an input, a list of chart I/Os of the type "IN", and if you have selected an output, a list of the type "OUT".

---

### Note:

Remember that the chart you have inserted by dragging it from the chart catalog to the current CFC chart is a copy. This nested chart no longer has any relationship to the original chart; in other words, modifications that you make in one of the charts are only available in that chart.

---

## Replacing

You can replace a nested chart with another (nested) chart. The interconnections of the "old chart" are retained in the new chart as far as possible.

To replace a nested chart, drag and drop the new chart from the charts catalog directly over the nested chart you want to replace. The chart is replaced when the mouse pointer was positioned within the chart you are replacing and you answered the prompt as to whether or not the chart should be replaced with "OK".

This "replace" function can, for example, be useful if you have configured nested charts as encapsulated functions and intend them for different uses. The following situation might arise:

You have encapsulated a (sub)function as a nested chart and interconnected the nested chart in a chart. The subfunction is, for example, a controller for a ventilation system of which there are different variants depending on where they are used. Within the overall structure, these variants can be interchanged without needing to change interconnections.

## 3.3 Handling Blocks

### Blocks for CFC

A basic pool of blocks is shipped with CFC. You can add further blocks from libraries or other projects.

### Help

The online help provides you with a description of the blocks shipped with CFC that you can display by clicking the block with the "Help arrow" (in the CFC chart, in the block folder or in the catalog) or by pressing F1 for the selected block.

---

**Note:**

**[S7]** The "CFC Library" contains the blocks for the CPU 3xxx in the "ELEM\_300" folder. These blocks cannot be run on a CPU 4xxx.

---

### 3.3.1 [S7] Importing Blocks

#### Importing into CFC

By importing, existing blocks are made known to CFC. You can only use such blocks in CFC.

You can import blocks by inserting them from the catalog (disguised import) or explicitly with the "Options > Block Types..." menu command.

Explicit importing is always useful when you require a lot of different blocks. Afterwards, inserting from the catalog is much faster since disguised importation is no longer necessary.

You would use the "Block Types..." dialog, for example when you want to import a number of blocks from the user program (block folder) into the chart folder. You would also use this dialog when 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 in the chart folder.

You can select the blocks you require in the "Offline block folder" box and then import the blocks to the chart folder by dragging them with the mouse or using the "-->" button. If a block already exists in the chart folder, a message is displayed.

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

## Number Conflict

If a block type is imported, and already exists with the same object name (block number, for example FB61) but with a different symbolic name, a dialog opens in which you can rename the block (same response as in the SIMATIC Manager when copying blocks with the same object name).

### 3.3.2 [S7] Importing a New Version

If, following a block type change, you want to use the modified blocks for the previously used old block types, the new block types must be imported into the chart folder. The block instances used in the CFC chart can then be adapted to the modified block types (central block type change).

Blocks are imported as follows:

- When you insert a modified block type from the S7 program or the block library into the chart.
- Using the "Options > Block Types..." menu command in the "Block Types" dialog. Here, you select the block to be imported from the list of source files ("Block Folder Offline") and then drag the block to the "Chart Folder" list (or click the "-->" button).

## Using a New Version

If a block type with this name but with a different version exists in the chart folder, the "Import New Version" dialog box with a message outlining the consequences and the version information of both blocks is displayed. If you answer the question "Do you still want to use the new version of the block types?" with "Yes" all the corresponding block instances will be adapted. If you answer with "No", no adaptation takes place. Three separate situations can be distinguished:

1. Changes that require neither downloading the entire program nor renewed AS - OS data transfer (compile OS), because they are relevant only within the ES (for example a block I/O was made invisible). In this case, you only need to download changes in RUN.
2. Changes that are relevant to the OS and therefore require an OS compilation (for example after changing a message text).
3. Changes in which the structure was modified, for example, adding I/Os and/or messages. This has the following consequences:
  - Interconnections and parameter settings can be lost.
  - The only option is to download the entire program; in other words, the CPU must be changed to STOP.
  - If the block is intended for operator control and monitoring (in other words, it will be executed on the OS), an OS compilation is also necessary.
  - If you want to retain parameter settings on the AS, the chart must be read back before downloading again.

---

**Note:**

The central block change is not made if the category of the block type has been changed (for example FB <--> FC).

The idea of central block type changes relates to FBs and FCs. The types of BOPs cannot be changed since they are an integral part of CFC.

---

### **Same Version**

If you import block types and the same version already exists in the chart folder, you will be informed that you do not need to import and will see a list of the identical block types.

### **Tidying Up Block Types**

With the "Tidy Up" button, you can remove blocks from the chart folder and from the block folder if they are no longer required in the project.

CFC checks whether there are unused block types in one of the folders. Depending on the result of the check, a dialog box is opened for each of the folders listing the unused blocks. (Block folder: Block types that were not imported into the CFC data storage system; Chart folder: Block types of which no instance blocks exist in the CFC charts). You can select the blocks to be deleted in the dialog box.

### **Blocks in Libraries**

If you have made a type change to blocks that were previously only in the library, please remember to include the blocks in the S7 program (block folder) first before you update the blocks in the chart folder with "New Version".

If you have made a type change to blocks within the S7 program (in the block folder), please remember that the library may also be updated.

### **Log of the Block Types**

The modifications resulting from a central type change are logged and automatically displayed following the update. You can also display the log again and, if required, print it out with the "Options > Logs...: Block Types" menu command. If modifications to block instances are necessary, you can use the log to help you saving both time and reducing the risk of errors.

### Tolerant Type Import after Data Type Extension

If the data type for a block type is extended, all data of the I/O are retained during a central type change if the old data type can be mapped on the new one without conversion. This applies to:

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

**Note:**

Remember that the existing interconnections to these I/Os are nevertheless lost as a result of the central type change.

### 3.3.3 Effects of Type Changes on Block Instances

If the interface description (block I/Os) or the system attributes of a block type are changed in the CFC chart, all instance blocks are automatically adapted. Depending on the change to the interface, this may have effects that require further action.

#### Effects of Changes on Block I/Os

- **An I/O is added**  
The change is also made in the block instances and the default is set for the system attributes. If there is no longer room at this position after increasing the size of a block, the block becomes an "overlapping block".
- **An I/O is deleted**  
The I/O is removed from the block instances. If the I/O is interconnected, the interconnection (or an SFC access) is also deleted. The deleted interconnection (or deleted SFC access) is entered in the log of the changes.
- **Order of the I/Os is modified**  
The order of the I/Os is taken into account and the interconnection, parameter assignment, and attributes are retained.
- **Data type of an I/O is modified**  
All the assigned parameters and interconnections of this I/O are lost. This situation occurs when you delete and then recreate an I/O.
- **Name of the I/O is modified**  
All the assigned parameters and interconnections of the I/O are lost. The reference to the old name cannot be established automatically by the system. This situation can occur when you delete and recreate an I/O.

## Effects of Changing System Attributes

If you modify attributes, the effects can be summed up in the general rule:  
What cannot be modified in the block instances is automatically modified in the instances as a result of a type change.

## Effects on Other Functions

- It is no longer possible to download changes only (only the entire program).
- The "Read Back Chart" function is no longer possible, since the blocks on the AS no longer match the configuration in CFC.

## [S7] Effects on WinCC

- If a block type is changed, DBs with new DB numbers can be created by compressing and compiling. To ensure online access is retained, the data must be transferred to WinCC again (OS compilation).
- If block I/Os intended for operator control and monitoring are affected (attribute S7\_m\_c=true), the following rules apply:
  - If an I/O is added, this is known in WinCC following the OS compilation and can be used in WinCC.
  - If an I/O is deleted, the WinCC tag no longer exists. The existing interconnection(s) must also be deleted in WinCC.
  - If the name of an I/O is changed, the name of the WinCC tag also changes. The interconnections of the picture elements, block icons and faceplates must be adapted.
- Whenever a change is made that affects WinCC, a new OS compilation is necessary.

### 3.3.4 Inserting Blocks in the Chart

#### Inserting in a Chart

You can insert blocks both in the sheet view and in the overview.

You can select a block type from other charts from the catalog of the blocks or libraries and drag it to your chart. These are copied automatically to the current S7 program and imported into the CFC chart. They then also appear in the block catalog (in a family of CFC blocks and in "All Blocks").

If you drag the block to a position where there is not enough space for the current block size, it is inserted as an "overlapping block". The block I/Os of overlapping blocks are not visible and the header and body are displayed in light gray. Once you move this block to a free position, it is displayed as normal.

If you want to insert blocks from the "S7 Program" of the block catalog, please remember the following:

These blocks in the S7 program are not yet known to 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.

Blocks that have already been imported are in one of the block families or, if the block is not assigned to a family (specified in the header), it will be in the "Other Blocks" folder. There are also listed in alphabetical order in the "All Blocks" folder.

---

**Note:**

Inserting blocks from the block families is the fastest way of placing blocks in your chart since no check is required. You should always select this method when a block type has already been imported.

---

#### Automatic Name Assignment

When you insert it in the chart, the block is assigned a number as its default name. This is continuous; in other words, each time you insert a new block (by copying or dragging from the catalog) the next higher free number is assigned.

If you change the default name, this applies to this block when copying or moving: If this results in two blocks with the same name, a number is appended (without brackets). If the last character of the modified name is a number, this is incremented.

Examples:

Block	<b>2</b>	copy	→	<b>3</b>
Block:	<b>REG</b>	copy	→	<b>REG1</b>
Block:	<b>REG3</b>	copy	→	<b>REG4</b>

## Specifying Run-Time Properties

When you insert a block, its run-time properties must also be specified, for example the point at which the block is inserted in the run sequence.

Default rule: The block is always inserted after the block displayed in the status bar.

You can change the installation position by selecting the block that will count as the "predecessor" for the next block to be installed and then selecting the menu command "Edit > Predecessor for Installation".

The current insert point is displayed to the right in the status bar. It displays the task name as well as the chart and block name after which the next block will be installed in the run sequence when a block is inserted in the CFC chart.

For more detailed information about run-time properties, refer to Section 3.7

## Aligning Blocks in the Chart

You can align the blocks inserted and selected in the chart, even those that are already interconnected, automatically (menu command: "Edit > Align > Left / Right / Top / Bottom"). Exception: This menu command does not work if overlapping blocks are selected.

The blocks are aligned at the visible edge of the block graphic (not the frame of the selected blocks). The block furthest to the left or right or nearest the top or bottom decides where the blocks are aligned.

If the alignment leads to a conflict, for example blocks overlapping, a message is displayed and the old situation retained. In this case, no blocks are aligned, not even those that normally could have been.

## Inserting and Editing a Text Box

You can drag a text box (icon:  New Text) from the block catalog and insert it at any position in the chart. You can enter text after you have opened the text box with a single click. You can adapt the size of the box to the text you have entered.

For more detailed information about editing the text box, refer to the online help.

### 3.3.5 Copying and Moving Blocks

You can copy or move blocks within a chart or to other charts. It is also possible to copy or move several or all the blocks of a chart together. This allows you to duplicate tested substructures quickly and without errors.

It is possible to copy blocks to a different CPU. The effects on the functionality are the same as those when copying charts to a different CPU.

The copied blocks are inserted in the other chart at the same position one grid to the right and down. At the same position means that the block is also inserted in the sheet with the same sheet number.

If objects are already located at this position, the blocks are displayed as overlapping blocks, otherwise they are displayed normally. You can then move the blocks to the required position (this is not absolutely necessary since even overlapping blocks can be executed on the AS).

In terms of the run-time properties and run sequence, the behavior when copying is the same as when inserting from the catalog; in other words, the insert point is determined by the "predecessor for installation".

If the blocks were installed in run-time groups, the run-time groups are not copied.

As far as possible, block names are retained. If there is a conflict, numbers are appended to the names.

Refer to the information in the online help.

### 3.3.6 Deleting Blocks

If you delete blocks, all interconnections between the deleted blocks and objects that you have not deleted are lost.

If an output is connected to a block that you are not deleting, a warning is displayed. If you decide to delete, instead of the capped interconnections, the inputs of the block that is not deleted are assigned the default parameter values.

## 3.4 Editing Blocks

Editing blocks in the chart means modifying their properties. Properties can relate both to an entire block or to individual inputs and outputs. The setting of properties of inputs and outputs is described in Section 3.5.

### 3.4.1 Setting Object Properties

The "Properties - Blocks" dialog box contains various tabs with which you can set properties. These are as follows:

#### "General" Tab

Here you can see all the properties that you assigned when you inserted the block (and the system-specific options in the "Special Object Properties" boxes).

The **Name** of the block is unique throughout the entire chart and is displayed in the block header (maximum 16 characters).

The **Comment** is a freely selectable text that appears in the block header. A maximum of 14 characters of this comment are displayed in the block header. In small blocks, the comment is not displayed.

Set the "**Operator C and M possible**" option when you intend the blocks to be used for operator control and monitoring on an OS. This activates the "Operator Control and Monitoring" button and the "Block icon" input box.

The "**Operator Control and Monitoring**" button opens a dialog in which the block I/Os that can be controlled and monitored are displayed with their WinCC attributes that can be edited here.

Blocks intended for operator control and monitoring can be displayed in WinCC with a **block icon** (with which a faceplate can be called). If there are different block icons available for a block type, so that different variants of a type can be distinguished (for example the MOTOR block as a motor, fan, pump etc.), these can be assigned to specific instances.

In the input field, you specify which block icon will be displayed in WinCC for this block.

With the "**Messages**" button, you open the dialog for configuring messages.

#### "Inputs/Outputs" Tab

You can assign parameters to block I/Os, add comments to them, include them in the watch list etc. see also Section 3.5.

### 3.4.2 Changing the Number of I/Os

If the block has a variable number of inputs and the same data type, for example NAND, OR etc., you can change the number of block inputs using the "Edit > Number of I/Os..." menu command.

The number of inputs is adapted according to what you select in the dialog box. If inputs are added, there may no longer be enough space for the block, in this case, it is displayed as an overlapping block that you can move within the chart.

## 3.5 Modifying the Properties of Inputs and Outputs

You can modify the properties of the block I/Os. Two different dialogs are available for this purpose:

- The dialog to edit all **I/Os** of a block is in the "Inputs/Outputs" tab of the "Properties - Block" dialog box that you can display by double-clicking the block. In the "Inputs/Outputs" tab, you will see all the properties of the I/Os in table form.
- For a **single I/O**, open the "Properties - Input/Output" dialog box by double-clicking the required block I/O. Depending on the properties and data type of the I/O, the dialog box will contain different information some of which can be edited and some of which is read-only.

### Note:

In the properties dialogs, you can enter units by editing directly or from a selection list. Depending on the area of application required by the user, these can be extended or modified as necessary (refer to the online help). By using the units in this list, you achieve uniform notation in the various applications (for example, process tag list, CFC Editor, SFC Editor) and avoid problems that can occur when using special characters.

### 3.5.1 Inverting a Block Input

Binary inputs can be inverted. After inversion, "0" becomes "1" and "1" becomes "0". This is only possible with interconnected inputs.

#### Setting the Inversion

You invert an interconnected binary input by selecting the input and clicking on the  icon in the toolbar or by selecting the menu commands "Edit > Invert Input" or "Invert" in the context-sensitive menu.

As an alternative, you can also double-click the block input to open the Properties dialog box and then set the "Inverted" check box.

#### Clearing an Inversion

To cancel the inversion, repeat the functions described above.

You can also cancel the inversion by deleting the interconnection at the input or moving it to another input or making a new connection to the input.

### Points to Note:

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

### 3.5.2 Value Identifiers

In addition to the identifiers "Text for 0" and "Text for 1" for the Boolean values 0 and 1, value identifiers are also possible for some numeric data types (BYTE; INT, DINT, WORD, DWORD).

#### Creating Value Identifiers

Using the value identifiers for the parameter values of the block or chart I/Os, you can define symbolic names. You make this definition with the system attributes ("S7\_string\_0" to "S7\_string\_25") when you create the block type or when you create the chart I/Os.

The text for "S7\_string\_0/1" can have a maximum of 16 characters and for "S7\_string\_2 to 25" it can have 8 characters. Only 8 characters are displayed. If you use more than 8 characters, you can decide which characters are displayed. You do this by including the "=" character in the text.

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

#### Displaying and Modifying in the Chart

With the data type BOOL, the value identifiers for "Text for 0" and "Text for 1" can be changed in each block instance if the system attribute is defined at the I/O of the block type. With all other data types, no individual modification in a block instance is possible.

In the chart, you can decide whether the symbolic value identifier or the absolute value is displayed. In the "Customize Layout" dialog box, you can activate or deactivate the "Parameter: Value identifier" ("Options > Customize > Layout...").

If value identifiers are defined for an I/O, you can select them in the properties dialog. The "Value" box then contains an additional button to open the drop-down list box. The selected value identifier is then displayed at the I/O.

## 3.6 Interconnections

In a CFC chart, an interconnection is the connection between

- an output of a block/chart and
  - one or more inputs of another or the same block or chart
  - an output of a chart
  - a run-time group (only BOOL data type)
  - objects outside the CFC data management (for example shared addresses)
- an input of a block/chart and
  - a chart I/O (within a chart)
  - objects outside the CFC data management (for example shared addresses)
- The data types of the input and output must be compatible. The interconnected blocks/charts can be in the same sheet, in different sheets of the same chart, or in other charts belonging to the same CPU.

One special form of a block interconnection is the textual interconnection with which the interconnection partner can also be located in a different chart folder or is not even known yet (see Section 3.6.3).

---

**Note:**

Each interconnectable input or output of a block is displayed with a connector pin in the chart. Inputs that cannot be interconnected because they have the attribute "S7\_link := false" can be recognized because the connector pin is missing. Such inputs cannot be interconnected either with the I/O of a block or nested chart nor with shared addresses. If, however, the CFC chart has chart I/Os, it is possible to connect such inputs to a chart I/O of its own chart.

---

You can also modify (rewire), copy, and delete these interconnections. By tracking signals (see 3.6.5) and using the sheet bar jumps, you can navigate through the chart.

Interconnections from one sheet to another or from one chart to another can be created easily by opening several windows at the same time and using the sheet view.

---

**Note:**

Certain interconnection rules apply to block I/Os of the following data types: ANY, STRING, STRUCT, and DATE\_AND\_TIME:

For more information, read the interconnection rules in the online help.

---

### 3.6.1 Interconnections to Shared Addresses

Shared addresses are connection endpoints located outside the CFC charts. The different Target systems recognize different shared addresses (for example in [S7] shared data blocks, I/O signals, memory bits, timers, counters).

With an I/O selected, you can display the address selection list with the menu command "Interconnection to Address". This contains all the symbols of the current symbol table that can be used for this I/O; in other words, that have a compatible data type.

For more detailed information about possible interconnections and examples of symbolic and absolute addresses, refer to the online help.

### 3.6.2 Interconnections to Run-Time Groups

You can disable and enable run-time groups dynamically. This means that an output value of a block decides whether a particular run-time group is executed or not. To achieve this, you connect the binary output of a block with the enable attribute of the run-time group.

#### Inserting

With the "Insert > Interconnection to Run-Time Group" menu command, you display a dialog box with a list of all the tasks on the CPU and a list with the corresponding run-time groups. After selecting the required run-time group (double-click), the interconnection is entered in the sheet bar.

#### Deleting

If you delete the run-time group, the interconnection is automatically deleted. The Enable attribute of the run-time group is set to "1" again.

To delete the run-time group, select the small field in the sheet bar and press "Del".

### 3.6.3 Textual Interconnections

A textual interconnection can only be located at a block/chart input and references a block or chart output in CFC. The textual interconnection remains an "open" interconnection that only becomes a "real" interconnection when it is closed.

A closable textual interconnection is the addressing of an input with a character string that identifies a concrete interconnection source (output).

#### Origin

A textual interconnection occurs

- automatically as a path reference (chart\block.I/O) with PH path (if it exists), when copying or moving to a different chart folder. When copying, the textual interconnection is produced at the input on the copy, and when moving both on the moved object and on the remaining object if the corresponding output is no longer in the same chart folder. The interconnection is deleted at the output if the input is no longer in the same chart folder.
- when entered by the user at the input (select the input and right-click or with the menu command Insert > Textual Interconnection)
  - as a path reference (chart\block.I/O) with PH path (if it exists). The interconnection is closed immediately if the interconnection partner is located in the same chart folder. If the interconnection partner is placed in the chart folder later, the interconnection can be closed using the menu command "Options > Make Textual Interconnection". A real block interconnection results.
  - as an interconnection request (any character string containing a concrete path reference, for example a comment). This interconnection request cannot be closed (warning in the log) and must be interconnected manually by the user.
- if there is a central type change if the data type of the interconnected I/Os no longer matches after the change.

**Please remember that** a textual interconnection **does not** result

- if the source of an interconnection is deleted. In this case, the real interconnection is deleted.
- with internal and connections to chart I/Os whose interconnection source (nested chart or block) is deleted or moved. In this case, the assignment to the chart I/O is lost.
- when there is a central type change involving a name change with interface change. Note: If there is a name change without an interface change, the interconnection remains.

## Display in the Sheet Bar

An **open textual interconnection** (interconnection partner not in the chart folder or the reference does not match any existing block I/O):

In the large field of the sheet bar you see the text you entered (one line).

In the small field, there is a yellow triangle as identifier.

By closing the textual interconnection, this is converted to a real interconnection; in other words, to a normal block interconnection.

The identifier in the small field (yellow triangle) disappears.

## Changing a Textual Interconnection

You can change existing textual interconnections as follows:

- Reconnecting to a different input with the same data type.
- Editing the text in the sheet bar. By double-clicking on the sheet bar (large or small field), you open the dialog box for entering the new text. The previous text is selected and can be overwritten.
- You can replace the textual interconnection with
  - an interconnection to the **shared address**: Select the input or the text in the sheet bar, right-click and select "Interconnection to Address...".
  - an interconnection to a **chart I/O**: Select the input or the text in the sheet bar, right-click and select "Interconnection to Chart I/O...". In the list in the dialog box, you will find all configured chart I/Os of the current chart and the I/O type (IN, OUT, INOUT).

## Closing Open Textual Interconnections

When all interconnection partners exist in the chart folder, you can convert these into real interconnections by closing the textual interconnections. There are two ways of doing this:

- With the menu command "Options > Make Textual Interconnection", you can close all textual interconnections of the current chart folder.  
Some textual interconnections cannot be closed.

A log is generated (Logs > Textual Interconnections tab) listing all closed and still open textual interconnections.

- Prior to compilation if the option "Make Textual Interconnections" is set in the "Compile Charts as Program" dialog box.

If one or more textual interconnections could not be closed, this is indicated in the log.

Compilation goes ahead and a substitute value (default value of the block type) is generated for the **input**. An interconnection at the **output** is ignored.

## Deleting and Finding Textual Interconnections

With the menu command **Options > Delete Textual Interconnections...**, you open a dialog box with the list of all textual connections.

In this dialog box, you can do the following:

- Obtain an overview of all existing textual interconnections of the current chart folder. With the "Cancel" button, you exit the dialog box without deleting the interconnections.
- Use the "Go To" button to display the chart in which the interconnection was set up. The interconnection is displayed flashing.
- Select textual interconnections and delete them from the chart folder with the "Delete" button. The interconnections are deleted immediately without any prompt for confirmation.

### General Information:

- A textual interconnection can be made up of a maximum of 512 printable characters.
- A textual interconnection can coexist with a closed interconnection or another textual interconnection.
- When a textual interconnection is closed, the actions and error messages are listed in the log and displayed. You can view the log at any time with the menu command "Options > Logs > Make Textual Interconnections tab".
- During compilation, open textual interconnections are tolerated; in other words, a warning is generated and the compiler follows the procedure below:
  - **Input:** The code is generated and the default value of the block type is used for the open interconnection.
  - **Output:** The interconnection is ignored in the generated code.

### 3.6.4 [S7] Interconnecting with SFC Charts (CFC in PCS 7)

#### SFC Access

A special form of interconnection is direct SFC access from actions or transitions to inputs or outputs (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 with 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.

- **Only move the SFC access:** Hold down the ALT key and drag the selected I/O to the required new I/O. The SFC access symbol is moved to the new I/O; any existing interconnection is not moved.
- **Move interconnection and SFC access:** Drag the selected I/O to the required new I/O. A message is displayed asking whether you want to move the SFC accesses as well. If you answer with "yes", the interconnection and the SFC access are moved, if you answer with "no" only the interconnection is moved.
- **Delete the SFC access:** You cannot delete an SFC access in the CFC chart.

In the CFC chart, a marker on the block I/O indicates these SFC accesses. Write and read access are indicated differently.

A marker above the I/O means "read access" and below the I/O means "write access". The color of the marker is the same as used to indicate the data type of the connection.

#### SFC References

You can display the SFC references in the "Properties - Inputs/Outputs" dialog. If the I/Os involve SFC access, this dialog box contains the "SFC Access..." button with which you can display a dialog box with the list of SFC references.

Double-click the name of a listed SFC chart to open the relevant chart where the referenced object is displayed as selected.

### 3.6.5 Handling Interconnections

#### Signal Tracking

Even when there are a lot of interconnections displayed in a sheet, you can nevertheless track the path of an individual connection.

Click the line or the large field in the sheet bar, all the lines and sheet bar entries, including those in overflow pages, that are connected to the selected line start to flash. You can track signals along branched lines and beyond connectors.

Starting at the chart I/O of a nested chart, the signal can be tracked to the I/O that is interconnected internally with this chart I/O. By right-clicking and selecting the "Track Signal" menu command, you open the nested chart; the interconnection including the sheet bar entry flashes.

To stop the flashing, simply click a free position in the chart.

#### Sheet Bar Jump

You can jump from a sheet bar entry of the sheet or overflow page to the interconnected block on the other sheet of the current chart or another chart:

- If you jump from an entry in the sheet bar or from **an I/O with a single interconnection**, the jump is made immediately.

The sheet or chart in which the connected block or nested chart is located appears. With block interconnections, signal tracking is activated (line flashes), with nested charts, 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 an entry in the sheet bar of **an output that has more than one interconnection**, a dialog opens with a list of all the interconnections of this output. The jump is made when you double-click on the required I/O (or select and click "OK").
- With the "Edit > Go To > Jump Back" menu command, you return to the point at which you started the jump even if you have closed the selected chart or have deleted the block.

#### Copying Interconnections

To copy an interconnection, select the connected input, press and hold down the Ctrl key and the left mouse button and then drag the mouse pointer to the required input (of the same type). When you have reached the required input, release the mouse button first and then the Ctrl key. A further connection is created.

Copying in this way is extremely convenient particularly when the interconnection originates in a block from a different chart, since you do not need to find the source.

## Modifying an Interconnection (Rewiring)

You can move existing interconnections from one input to another input or from one output to another output.

- Input:

Position the mouse pointer on the input whose interconnection you want to move to a different input and hold down the mouse button. Then drag the mouse pointer to the other input (of the same type) and release the button. The connection line is redrawn.

- Output:

- If you have multiple interconnections, they will all be moved to this output.
- If you drag the interconnection with an address to an output that is already connected to the same address, a warning is displayed. You can decide whether to retain the interconnection with the original block or delete it.

If you want to rewire an interconnection that is also accessed by an SFC chart, you will be asked if you want to move the SFC access as well.

## Deleting Interconnections

You delete an interconnection as follows:

- Multiple interconnections:

- If you want to delete **all the interconnections** of an output, select the output or the connection line(s) and select the "Edit > Delete" menu command or press the DEL key.
- If you want to delete **one of several interconnections** of an output, select the input and press the DEL key  
*or*  
position the mouse pointer on the sheet bar entry of the output and click the right mouse button. The small field before the sheet bar entry is selected. Select "Delete Interconnection(s)" in the context-sensitive menu and the interconnection to the input will be deleted. You can, of course, also click the small field in the sheet bar directly and delete the relevant interconnection with DEL or with "Edit > Delete".

- Single interconnection:

You can select the output, the input or the connecting line and delete the interconnection with the "Edit > Delete" menu command or with the DEL key.

Note: An SFC access cannot be deleted in an SFC chart.

### 3.6.6 Structures

Using a structure, any data can be put together in a tree structure and then applied to a block I/O. In the block display, a structure has an I/O name and a type; the type is "ST".

A structure is made up of several elements (with a nesting depth of up to 8 levels):

- Elementary data type (BOOL, WORD, .. )
- Structure

A structure element contains the following information:

- Type (of the elementary data type, otherwise "ST")
- The name
- The value (only with the elementary data type)

#### Interconnecting

A block I/O with a structure can only be interconnected with another structure not with an elementary data type. The structures you interconnect must be compatible; in other words, the order, the data type, and the element name of the elementary data types contained must be identical. The names of the structures can be different.

The elements within a structure cannot be interconnected separately, only the block I/O can be interconnected as an entire structure.

For more detailed information, refer to the online help.

#### Editing the Elements of the Structure

You can display the object properties of the structure or of an elementary data type in the structure and edit it (assign parameters).

Double-click the block I/O to display the "Select Structure Element" dialog. In the structure that appears, select the element you want to edit and open the Properties dialog with the "Properties" button (or by double-clicking the element). You can display the dialog box both in the Edit and in the Test mode.

If you want to watch a structure in the test mode, you can include individual structure elements in the dynamic display by opening the "Select Structure Element" dialog box (context-sensitive menu "Include in Dynamic Display").

#### Note:

- In structures, the system attributes for "operator control and monitoring" or "message" must not be used at any point.
- Structures cannot be edited in SFC.
- A block I/O with a structure cannot be interconnected with a shared address that refers to an entire data block (DBx); interconnections to structures within the DB are possible (DBx.name\_st).

### 3.7 Run-Time Properties of the Blocks

The run-time properties of a block decide how the block is included in the run sequence within the entire structure of the CPU. These properties are decisive for the response of the Target system in terms of reaction times, dead times, or the stability of time-dependent structures, for example closed loops.

When it is inserted, each block is assigned default run-time properties by installing it in the run sequence in a task (see also: Modifying the Run Sequence, Section 3.7.2). Where applicable, blocks can also be installed in run-time groups that are themselves installed in tasks.

**Note:** When you create a new chart, a run-time group is created automatically in which all the blocks of this chart will be installed.

Run-time groups are used to structure or organize tasks. The blocks are installed sequentially in the run-time groups and can be given the attributes for "scan rate" and "phase offset" in the object properties dialog.

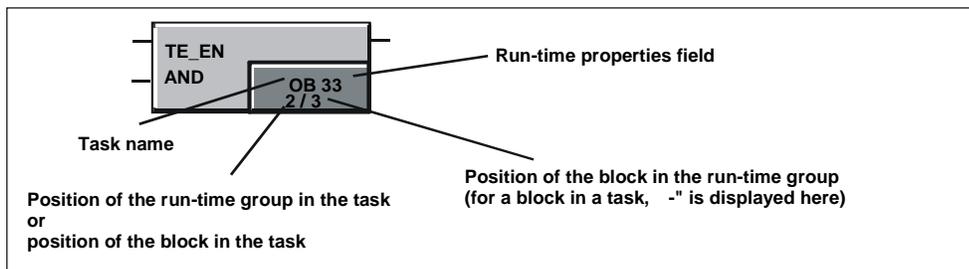
Each block must be installed in the run sequence at least once. A block can be installed in several tasks, however only once per task.

#### Displaying Run-Time Properties

There are several ways in which you can display the run-time properties; either for an individual block or for the entire CPU.

#### Single Block

The run-time properties of each block are displayed in the part of the block header on a colored background.



## Entire CPU

You can obtain a complete view of the run sequence as follows:

- By double-clicking the run-time properties box in the block header.  
The run-time editor is started and the task in which the block is installed (first point of installation) is opened in the run sequence window. The relevant block is selected.
- By selecting the menu command "Edit > Run Sequence..." or clicking the  button.  
The run-time editor is started and the window with the run sequence is displayed. If a block was selected, the task in which the block is installed (first installation point) in the run sequence is opened. The relevant block is selected.  
In this window, you can edit the run sequence of the entire CPU (see Section 3.7.2).
- By selecting the "Options > Chart Reference Data..." menu command or the  button.  
A separate application with its own window and menu commands/toolbar buttons is started.  
In the "Run Sequence" view (activated with "View > Run Sequence" or by clicking the  button), the complete run sequence of the current CPU is displayed (blocks with information on block type and comment). The tasks in which objects are installed are opened. You can print out the view of the run sequence as a list.  
In this window, you cannot edit the run sequence.

### 3.7.1 Sequence Model of CFC

#### The Concept of the Sequence Model

The sequence model allows a chart-oriented structuring of the run sequence; in other words, the blocks of a chart are installed one after the other in the run sequence. The installation pointer decides the next insert point for installation in the run sequence. There is both a chart installation pointer and a block installation pointer.

In an S7 program, there is one (global) chart installation pointer and one (local) block installation pointer for each chart.

In conjunction with the run-time group created automatically for each chart, the concept of the installation pointer ensures better chart orientation. Among other things, this allows the blocks of a chart to be handled as groups when branching and merging a project in multiuser engineering. This allows you to work on a chart-by-chart basis and to decide the run sequence of the blocks regardless of the run sequence of other charts.

If the installation pointers are not changed by the user, installation is as follows:

- New charts are installed after the existing charts
- Blocks are installed within a chart one immediately after the other.

Further advantages:

- The insert point cannot be moved accidentally in chart-oriented multiuser engineering by other configuration engineers.
- When modifying only one chart, you do not need to compile and download the entire OB but only the run-time group of the modified chart.
- The CPU utilization can be influenced on a chart-by-chart basis.
- Processing of individual charts by the CPU can be activated and deactivated (even in the test mode) without requiring an external controller (for example SFC).

### Chart Installation Pointer

Each program has its own global "chart installation pointer". The chart installation pointer decides

- the **OB** in which the run-time group of the next chart created will be installed.
- the **position** after which run-time group (or after which block at the OB level) the run-time group of the next created chart will be installed. If the OB does not yet contain a run-time group, this position is "empty"; in other words, it points to the OB itself.

The default of the chart installation pointer is OB35. The chart installation pointer is set only in the run-time editor (not in the CFC Editor).

See also "Setting the Installation Pointer" in Section 3.7.2

### Block Installation Pointer

Every CFC chart (including nested charts) has its own "block installation pointer". The block installation pointer decides

- the **OB** in which the next block to be inserted will be installed (main installation).
- the **position** after which block the next new block will be entered in the run sequence. The block position is always linked uniquely to the OB number.

If the chart is empty, the block installation pointer is "undefined"; in the status bar, the chart installation pointer is displayed instead. When the first block is inserted, the pointer is taken from the current chart installation pointer.

The block installation pointer is set only in the CFC Editor (not in the run-time editor). The default (after creating a new chart) is the start at the beginning of the automatically created chart run-time group.

See also "Setting the Installation Pointer" in Section 3.7.2

## 3.7.2 Modifying the Run Sequence and the Installation Pointer

### Starting the Run-Time Editor

With the menu command "Edit > Run Sequence..." or by clicking the  icon in the toolbar, you open a divided window with the hierarchy window on the left and a detailed window on the right. This is structured similar to the Windows Explorer and working with it is similar.

### Moving Objects

You can move an object (chart, run-time group or block) by selecting it (in the right or left window) and dragging it to the object after which you want to install it (drag-and-drop).

If you drag an object to a run-time group,

- the object is installed at the first position within the run-time group when the structure is expanded [-].
- the object is installed after the run-time group if the structure is not expanded [+].
- and the run-time group is empty, you will be asked whether or not you want to install the block within the run-time group. If you acknowledge with "yes" the object will be installed **inside**, if you answer "no", it will be installed **after** the run-time group.

If you drag an object to a block/chart within the run-time group, the object will be installed after this block/chart.

If you drag an object to a task, it is installed **before** the existing installed objects.

**Note:** When moving blocks, make sure that all blocks of a chart are located only in the relevant run-time group. After moving a block to another group, the chart-oriented structure no longer exists and would make it difficult or even impossible to work on a chart-by-chart basis in multiuser engineering.

### Removing a Block

You can only remove blocks (delete) from a task if they are installed more than once in the run sequence. The block must remain installed at least once.

If the block is installed only once, you cannot delete it. If it exists more than once, the block is deleted and the run sequence of the blocks following this position is updated.

## Installing Blocks

You can install blocks, run-time groups, and SFC charts more than once using Copy and Paste. You use the relevant menu commands or the icons in the toolbar or drag the objects while holding down the CTRL key (see also: Moving Objects).

You can also install blocks by dragging them directly from a CFC chart to the required position in the run sequence (when both windows are displayed at the same time).

### Note:

Objects with the system identifier "@" are installed automatically in the run sequence when the block drivers are generated and should only be edited with the SIMATIC Manager function "Options > Charts > Generate Module Drivers..."; in other words, they should not be moved or deleted manually.

## Setting Installation Pointers

You can modify the installation pointers as follows:

- **Chart Installation Pointer** (Default OB35)  
To modify the chart installation pointer, open the **run-time editor** and select the required OB or a block at the OB level (not within a run-time group) or a run-time group within the OB.  
In the run-time editor, select the menu command "Edit > Predecessor for Installation".
- **Block Installation Pointer**  
You cannot set the block installation pointer in the run-time editor. To modify the block installation pointer, open the **CFC Editor** and select the block after which all other blocks will be inserted.  
In the chart, select the menu command "Edit > Predecessor for Installation".  
  
If the block selected as the predecessor for installation is deleted, the block installation pointer is decremented; in other words, set to the block installed before the deleted block. This also applies if the block is moved to a different chart. The block installation pointer in the destination chart is not changed. The moved block retains the installation position it had in the previous chart.

For more detailed information on working with the run-time editor, refer to the online help.

### 3.7.3 Run-Time Groups

#### Applications

Run-time groups  are optional but are created automatically when you create a new chart. They are used to structure tasks (OBs). The blocks are installed sequentially in the run-time groups.

With the run-time groups, you can do the following:

- Remove or reconnect selected blocks for execution on the AS.
- Execute selected blocks with a specific scan rate (every nth number of cycles) and/or with a phase offset to achieve better load balance on the CPU.
- If OBs contain a large number of installed blocks, these can be put together in smaller units.

**Advantage:** Instead of creating one "large" FC when you compile each OB, "smaller" FCs are created depending on the number of run-time groups.

If the program is modified later, only the run-time groups/FCs that actually contain modified blocks are given the "modified" ID.

This means that later delta compilations and downloads of changes take far less time.

- When branching and merging a project in multiuser engineering, this allows the blocks of the chart to be handled as groups. This allows you to work on a chart-by-chart basis and to decide the run sequence of the blocks regardless of the run sequence of other charts.

#### How Are Run-time Groups Created?

Run-time groups are created in two ways:

- **Automatic**

When you create a new chart, a new run-time group is created automatically and installed in the run sequence according to the chart installation pointer. This automatically created run-time group has different properties from a run-time group created manually:

The automatically created run-time group and the chart have a certain dependency on each other that remains until modifications are made to the run-time group. Modifications include, for example, name changes and/or subsequent installation of other blocks in this run-time group.

This dependency means that the name of the run-time group is identical to the chart name and if the chart is renamed, the run-time group is automatically renamed with it. If the chart is deleted, the run-time group is also deleted if it becomes empty as a result of deleting the chart.

If this dependency is no longer present (because changes have been made to the run-time group) the automatically generated run-time group behaves in just the same way as a manually inserted run-time group.

- **Manual**

You can create run-time groups manually at any time (create new groups with a menu command or copy existing groups). If you copy a run-time group and insert it in the run sequence, a continuous number is appended to the name (without brackets).

You create a run-time group by selecting the task in the left-hand window in which the run-time group will be installed or the object following which the run-time group will be installed. With the "Insert > Run-Time Group..." menu command, you start the "Insert Run-Time Group" dialog. Here, you can specify the properties of the run-time group.

### Editing and Deleting Run-Time Groups

You can also edit run-time group properties later.

To **edit**, select the run-time group (in the left or right window) and then select the "Edit > Object Properties..." menu command.

In the "Properties - Run-time Group" dialog box, you can modify the existing values (name, comment, phase offset, scan rate).

The options "Optimization of the run sequence" (see Section 3.7.4) and "Active" are defaults (set). With "Active", the run-time group is activated for processing in the test mode (option set) or deactivated (option reset). Note: In F-systems, the option is always set and cannot be modified.

You can **delete** run-time groups only when they contain no objects. To delete a run-time group, select it and then select the menu command "Edit > Delete" or press the DEL key.

If the deleted run-time group was selected as the "Predecessor for Installation", the default insert position becomes the next element of the task before the deleted group, or if no further element exists, the start of the task.

### 3.7.4 Optimization of the Run Sequence

This function optimizes the run sequence of a program according to the data flow so that as few dead times as possible occur on the CPU while the program is running. The optimization is separate for OBs/tasks and run-time groups.

You start the optimization in the run-time editor using the menu command **Options > Optimize Run Sequence**.

#### What Happens during Optimization?

This is handled separately for each task. Within a task, the run-time groups are handled extra. The scan rate and phase offset of a run-time group are ignored.

The data flow is obtained from the interconnections. These include all block-block interconnections as well as those to SFC charts and interconnections of block outputs to ENABLE a run-time group. Global and textual interconnections are not taken into account. Interconnections to the chart interface are tracked as far as the actual source of the interconnection. If this source does not exist; in other words, the interconnection ends at an interface, the interconnection is ignored. Interconnections to blocks located in other tasks are also ignored as is access from SFC charts to block I/Os.

Interconnections into a run-time group or out of a run-time group are considered to be interconnections of the run-time group itself; in other words, a run-time group forms a fictitious block at the task level. Interconnections between the blocks of a run-time group are used only for optimization within the run-time group. This ensures that the run-time groups are correctly arranged in themselves and that the run-time group itself is placed at the optimum position within the task.

Subsequent optimizations are designed so that no unnecessary changes are made and that the scope of changes is kept as limited as possible when changes are compiled and downloaded.

## 3.8 [S7] Generate Module Drivers

To allow signal processing in PCS 7, a function is available that automatically generates the required module drivers and interconnects them after you have configured the hardware with HW Config and configured the technological functions in CFC. This function, also known as "driver generator" in the ES can be used as follows:

- You can set the option in CFC in the "Compile Charts as Program" dialog box. The module drivers are then generated before compilation.
- With the chart folder selected, you can activate the function in the SIMATIC Manager with the menu command "Options > Charts > Generate Module Drivers...".

The driver and message concept described here can only be used for CPUs of the S7-400.

### Driver and Message Concept

For this functionality, blocks types are used that have the effect of separating the hardware and software configuration:

- The channel-specific blocks are inserted in the chart and interconnected by the user with appropriate signal names from the symbol table. These signal preprocessing blocks are part of the technological function and are always processed along with it.

The following types are available for signal preprocessing:

- **Standard channel blocks:**CH\_AI, CH\_AO, CH\_DI, CH\_DO. These are used only for signal processing of S7-300/400 SM modules. You use these standard blocks when you want to achieve a memory and run-time optimization and no PA devices need to be processed.
- **Universal channel blocks:**CH\_U\_AI, CH\_U\_AO, CH\_U\_DI, CH\_U\_DO. These blocks are used for signal processing of S7-300/400 SM modules or a PA field device. The advantage of these blocks is that you can create CFC charts that are independent of the hardware I/O used later. The disadvantage is that the universal blocks require more memory and take up more run time.
- **PA channel blocks:**PA\_AI, PA\_AO, PA\_DI, PA\_DO, PA\_TOT. These blocks are specially intended for use with the PA field devices. They are used mainly when the special properties of these devices are required. In contrast to the CH blocks, not only the signal itself is processed, but all variables according to the selected desired configuration of the device in the defined hardware configuration.
- The blocks for diagnosis and monitoring of I/O modules, field devices, links and for signaling and displaying CPU events and states are generated automatically by the system (inserted in the system charts from the block library by the driver generator), assigned parameter values, and interconnected.

For more detailed information on the functions, functionality, and message capability of the CH blocks, refer to the context-sensitive help (F1) for the block.

## Use and Selection

With CFC Version V5.2 + SP1, an extended driver concept was used in which control of the driver generator was achieved with metaknowledge. The driver generator can be executed according to the previous or according to the extended concept. Which concept is used for existing projects is decided based on the signal processing blocks used (CH blocks). The driver generator searches through the imported block types and checks the version of the CH blocks. The previous concept is then used if the CH blocks have a version less than 2.0 .

## The Extended Driver Concept

With the extended driver concept, more modules are supported for ET200M, ET200S, ET200X, PA devices, DP/PA links, DP/PA-Link, Y-Link, DP V0- / DP V1 slaves, diagnostic repeaters, HART field devices (diagnostics) and for redundant I/O released for PCS 7 and exact specification of channel errors is possible in messages. Adopting configured data from SIMATIC PDM is possible.

With a new hierarchical interconnection model of the driver blocks in conjunction with the use of the new blocks OB\_BEGIN and OB\_END for diagnostics on the CPU and connection (instead of MSG\_CSF), a run time optimization is achieved.

When extending hardware that requires its own diagnostic and signaling concept and its own blocks for processing, this metaknowledge must also be extended in the form of XML files. **At the same time, existing XML files must not be modified.**

The basic set of XML files (object lists and action lists) is maintained with the PCS 7 Library and installed with Setup. The corresponding folders "Object" and "Action" are located in the path: ...**Siemens\Step7\S7data\driver\...**

## Session Model for "Generate Module Drivers"

- The user configures the **hardware** with the symbolic names of the I/O signals. The signal names are entered directly in the symbol table by HW Config (select module, menu command in HW Config: "Edit > Symbols...").
- The user configures the technological functions in CFC, Here, the CH\_ blocks (or PA\_ blocks) are used and they are interconnected with the signal names from the symbol table ("Insert > Interconnection to Address...").

- The user starts the driver generator that creates, interconnects and assigns parameters to all module drivers in the SIMATIC Manager with the menu command "Options > Charts > Generate Module Drivers...". The following phases are run through automatically:
  - All the channel blocks (\_AI, \_AO, \_DI, \_DO) used in the CFC charts are identified. Then all the I/Os are detected (for example Value, O\_SP, I\_OUT\_D, ...) that are interconnected with the symbols of the input/output channels configured in HW Config. With the PA\_AO-/PA\_DO blocks this can also be several symbols that need to be interconnected. Here, only a "leading" symbol needs to be interconnected, the others are identified and interconnected automatically by the "Generate Module Drivers" function. Using the addresses of the symbols in the symbol table, the corresponding module and the channel of the module are identified from the hardware configuration data. Based on the module type, the corresponding diagnostic block type (MOD or PADP) is obtained and a block instance then created in the system chart for each I/O module.
  - The parameters required for the diagnostic blocks are obtained from the HW Config data and entered in the block instances (also in the SUBNET block for optimizing the run time in the error OBs).
  - The interconnection between the channel-specific output on the MOD/PADB block and the corresponding input on the CH block is created. If the module supports "value status", the address of the value status is obtained and interconnected with the CH/PA block.

If changes are made in the configuration (hardware or software), the changes must be processed using the driver generator. The existing driver blocks are then not deleted and recreated but simply have parameters reassigned. Blocks that are no longer required are deleted (except for those included by the user) and additional required blocks are created. If block types have already been imported those from the CFC data management are used and not those from the library.

## How "Generate Module Drivers" Works

System charts are generated automatically and all the necessary diagnostic blocks are inserted in them. The system charts have the names "@1", "@2", etc.

---

### Note:

Objects identified by the "@" character should not be modified by the user, but should only be manipulated using the "Generate Module Drivers" function. In the same way, diagnostic blocks should not be inserted manually.

Driver blocks created by the user must be inserted from the block catalog. If "@" blocks are copied, these are deleted the next time the "Generate Module Drivers" function is executed.

---

## New Block Types

If a **new PCS 7 library** with modified block types is installed, initially, this has no effects on blocks already in use.

If, however, you want to update the blocks, follow the steps outlined below:

- Delete all @ system charts in the chart folder.
- Select the "Options > Block Types..." menu command and delete the relevant blocks in the "Chart Folder" window (select the blocks and click the "Tidy Up" button).
- In the component view of the SIMATIC Manager, start the "Generate Module Drivers" function.  
New system charts are then created and the driver blocks from the new library are used since they now no longer exist in CFC (the library is entered in the "Settings" dialog box).

## 3.9 [S7] Compiling

You can compile the chart (in other words all the charts of the chart folder) as a program or an individual chart as a block type.

### 3.9.1 [S7] Compile Charts as Program

During compilation, the charts of the active CPU are converted to machine code. In S7, the SCL Compiler is used.

You start compilation with the menu command "Chart > Compile > Charts as

Program..." or the  icon in the toolbar.

### Relationships

If you want to control the assignment of resources on the CPU using the "Customize Compilation" function, it is important understand the relationships explained below.

The structure you configure in CFC is mapped to the S7 machine 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 run-time group

DBs are used as follows:

- An instance DB is created for each instance of an FB.
- To store interim results for example from FCs, CFC-internal DBs are created. One of these DBs is required per data type. When the maximum length (4 Kbytes) is reached, a further DB is created.

### What is Compiled?

When you compile, you can choose between two options:

- **Range: Entire program**  
If you compile the entire program, the content of the entire chart folder is compiled regardless of whether changes have been made or not.
- **Range: Changes**  
If you compile the changes, only the objects that have changed since the last compilation are compiled. Changes are solely changes in content (interconnections, block attributes, number of inputs etc.), but not the movement of objects within the chart. By selecting "changes only" the time required for compilation is significantly shorter and this should be the setting you normally choose.

**Note:** When you compile the changes, only the changes are checked. If you want to check the entire program, start the consistency check.

## Optional Functions Prior to Compiling

Before you compile you have the opportunity to select certain options so that the functions execute before the actual compilation starts. This includes the following:

- "Generate Module Drivers" Option  
If this option is set, the module drivers for the existing signal-processing blocks are generated and connected to them prior to compilation.
- "Update sampling time" Option  
If blocks have an I/O for the sampling time (in PCS 7 blocks, the input SAMPLE\_T, attribute "S7\_sampletime = true"), the sampling times can be updated.  
  
If you set the "Update the sampling time" option (in the compilation dialog), the system checks which cyclic interrupt OB the block is installed in before compiling. The cycle time of the OB is then written to this input. If the block is installed in a run-time group, the scan rate is also taken into account when the sampling time is calculated.
- "Delete empty run-time groups" Option  
If this option is set, empty run-time groups are deleted prior to compilation.  
  
These empty run-time groups can result from copying when branching and merging project data.
- Option: "Make Textual Interconnections"  
If this option is set, all the textual interconnections are closed prior to compilation if the referenced interconnection partner exists; in other words, they are converted to real interconnections.  
  
**Note:** If the option is not set or if there are textual interconnections that cannot be closed, substitute values are generated; in other words, the default parameter value of this block type is used.

## Consistency Check

During compilation, a consistency check is made automatically. You can also start this check manually, for example before you overwrite an old executable CFC program by compiling.

## Preparing for Compilation

With the "Options > Customize > Compilation..." menu command, you can call a dialog box in which you can reserve FC and DB numbers for your own use. These areas are not used by CFC (areas reserved for other applications).

For explanations of the dialog and further information about the settings, refer to the online help.

### After Compiling

When the compilation is completed (or aborted), the "Logs" dialog box is displayed. The entire compilation is logged here including warnings and errors. Based on the log, you can check whether compilation was correct and can print out the log with the "Print" button.

You can also display the log again and, if required, print it out with the "Options > Logs..." menu command.

---

#### **Note:**

Compiling the entire program does not necessarily mean a complete download. If the program was already loaded on the CPU prior to compiling, it is possible to download the changes only.

---

### 3.9.2 [S7] Compile Chart as Block Type

You can create block types from existing CFC charts that you want to use more than once. You can assign system attributes to these block types.

You start the compiler with the "Chart > Compile > Charts as Block Type..." menu command. A dialog box with two tabs opens.

In the "General" tab, you can specify the properties of the block type before you compile (FB number, symbolic name, name (header), family, author, and version (header)) and specify the Target system on which the block will be used (S7-300 or S7-400).

You can optimize the code. For the optimization, you can choose between "local data requirements" and "download changes in RUN":

- **Local 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 are stored in the VAR\_TEMP area, as far as possible. As far as possible means that all interim results in the data flow are stored here. Only the interim results that are not in the data flow (for example in feedback loops) continue to be stored in the VAR area (instance DB). The advantage of this optimization is that not all changes lead to a change in the interface time stamp of the instance DB, so that in most cases the changes can be downloaded online. One disadvantage is that the local data requirements are increased.

As an option, you can also activate the know-how protection. As a result of this, the algorithm of the block can only be seen and, if required, modified when the suitable SCL source files exist.

In the "Attributes" tab, you can enter the system attributes for the block type (for example the system attribute "S7\_blockview" value "big").

For further information, refer to the online help.

## 3.10 Downloading the User Program to the Target System

To test a user program for a CPU and to start it up, it must be downloaded to the Target system.

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

---

### Note:

With the programs created in CFC, you must always download to the Target system from CFC, since only the download function of CFC guarantees the consistency of the configuration data with the AS data.

The same download function is also used if you select the "AS >

Compile/Download Objects" menu command in the SIMATIC Manager.

On the other hand, copying blocks in the "Offline Block Folder" to insert them in the "Online Block Folder" is not permitted.

---

### Downloading

To download the user program, select the "PLC > Download" menu command (or click the  button in the toolbar).

A dialog box is then displayed in which you select the type of download and start a further dialog for options.

If download-relevant changes have been made to the program, a message is displayed indicating that the program must first be compiled and you are asked whether you want to compile and then download.

### Notes:

- Compiling the entire program does not necessarily mean a complete download. If the program was already loaded on the CPU prior to compiling, it is possible to download the changes only.
- If a full download is aborted, no download of changes is possible until the full download is completed. Reason: The blocks were deleted on the CPU prior to the download.
- You can compile a program as often as you wish (changes only or entire program) without losing the ability to download changes.

## [S7] Download

You can download a program in one of two ways: "Entire program" or "Changes only". If you select "Entire program", all loadable objects of the program are downloaded; if you select "Changes only", you download only the changes since the last download.

The "entire program" can be downloaded in the STOP or RUN-P modes. In the RUN-P mode, the CPU is set to STOP after a prompt for confirmation and all the blocks it contains are cleared. After successful downloading, the message "Do you want to restart the CPU?" appears; answer with "Yes" to trigger a warm restart.

You can download "changes" in the "RUN-P" CPU mode. The correct download order with the granularity required for the data to be downloaded is maintained. Please note that there is no absolute guarantee that the CPU will not change to the STOP mode due to inconsistencies.

The system runs a wide range of checks that prevent a download if an error is detected. Refer also to the CFC online help "Downloading the User Program to the AS", topic "System Support for Avoiding Causes of CPU Stop".

Notes on downloading changes:

- It is basically always possible to download changes when
  - a full download has been done at least once.
  - you download the program structure that was last downloaded.
- Each download involves an asynchronous compression of the CPU. This avoids warnings and error messages due to lack of storage when changes are downloaded later. Compressing the CPU storage 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 a complete download. If the program was already loaded on the CPU prior to compiling, it is possible to download the changes only.
- You can compile a program as often as you wish (changes only or entire program) without losing the ability to download changes.
- If a changes download was aborted, the remaining parts can be downloaded in a renewed attempt.

**Note on H CPUs:** If the H CPU is in the solo mode, for example after the failure of a CPU and there was a CPU failover, an online access results in a selection dialog being displayed. In this dialog, you can select the required CPU. In the redundant mode, this dialog does not appear.

**Note on F systems:** You can only download changes for programs with modified F components after entering an F password. Without this legitimization, downloading is aborted.



### Caution!

Further causes of a stop are listed in the online help topic "Reasons for STOP when Downloading Changes Online". Please read this topic before you download changes online to the CPU in the "RUN-P" mode.

---

### **Option: Include user data blocks**

If this option is set (default setting) when you download changes, the data blocks that are not in the CFC area are handled as follows:

- They are included in the download if the time stamp is different or data blocks have been added.
- They are deleted on the CPU if they do not exist in the S7 program.

If you want the user data blocks to be ignored when downloading changes, deactivate the option.

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; this is also the case if the user data block in the S7 program differs from the block on the CPU. Correct execution of the program is then the user's responsibility.

### **Saving Settings without Downloading**

You can save the settings in the dialog box without starting the download ("Apply" button). This can, for example, be useful if you want to use the "Compile and Download Objects" function in the SIMATIC Manager.

## 3.11 [S7] Reading Back Charts

### Modified Data

You can save (read back) the parameters of all CFC charts contained in the chart folder of the active chart and whose AS data were changed (for example limit values or control parameters changed by operator input from an OS with WinCC).

### Reading Back

You start the read back function with the "Chart > Read Back...." menu command. If you select "Program of the CPU" as the source in the "Read back" dialog box, the program on the CPU (online block folder) is read back to the chart folder with the current parameters.

Another method of reading back when you previously read out the data from the CPU using the SIMATIC Manager (with the "AS > Upload Station" menu command) is to specify "Program Offline" as the source in the "Read Back" dialog box. The data are then read from the offline block folder and written to the chart folder.

In either case, in the dialog you can select whether you want to read back all parameters of the block inputs or only the parameters assigned the system attribute for "operator control and monitoring" (S7\_m\_c:=true').

After reading back, there is automatically a compilation of the entire program. This does not, however, force a download of the entire program, this can still be downloaded in RUN to the CPU (changes only download).

A log is created and displayed on completion of the read back. You can display the log later with "Options > Logs > Read Back" tab. Among other things, the log displays which I/Os were read back and modified and where problems occurred.

### Notes on Reading Back

- I/Os that cannot be assigned parameter values (Attribute: S7\_param := 'false') cannot be read back.
- Values of FC inputs and the data types ANY, ARRAY, POINTER, UDT are ignored during the read back.
- If values are modified on the AS and the chart is then read back, these modified values are marked as downloaded, in other words if you then download changes to the CPU, these values are not included.
- Certain blocks can be excluded from the read back function (for example BATCH blocks). In this case, the block type has the attribute S7\_read\_back := 'false'. The attribute cannot be modified in the block instances.
- **Note on H CPUs:** If the H CPU is in the solo mode, for example after the failure of a CPU and there was a CPU failover, an online access results in a selection dialog being displayed. In this dialog, you can select the required CPU. In the redundant mode, this dialog does not appear.

## 3.12 Several Engineers Working on one Project

It is always possible for more than one engineer to work on one project or multiproject. This allows configuration, testing, and commissioning of ASs to be performed at different locations or in a PC network (multi-user mode).

### Basic Options

- **Networked PCs.**  
The project is located on the server and several engineers configure defined parts of the project at the same time (AS, OS). Make sure that the charts of a AS or the pictures of an OS are edited on **only one PC** (by one user) at **any one time**.
- **Projects within a Multiproject.**  
The projects of a multiproject are stored centrally on one computer and can be distributed for editing on other computers. To edit the individual stations, the projects are transferred to the workstations of the engineers involved. Several projects can be present on the one workstation computer at the same time.  
  
After editing and merging back into the multiproject, the data that are relevant beyond a single project must be synchronized with system support (menu command "File > Multiproject > Adjust Projects") and the relevant interproject functions must be started (for example, compile OS). This function should be run at the latest when the configuration is being prepared for commissioning.
- **Distributing and Merging Project Data.**  
You can distribute the charts of an S7 program in several (branch) projects. Distributing individual charts of a project and merging them again after separate editing is possible with the new run-sequence model of V6.0 that allows chart-oriented structuring of the run sequence.  
See also: Run Sequence Model of CFC, Section 1 and Textual Interconnections When Branching and Merging Project Data, Section 3.12.1

### How to Branch and Merge Project Data

1. Copy a technological part of the project (single chart, several charts) to a different project.  
Result: The copy contains textual interconnections to all sources that were not copied.
2. Edit the copied section separately (add, delete, modify blocks and charts).

3. Copy the edited technological part of the project back to the original project.  
Result: The system first deletes the charts with the same names in the original project. There are now textual interconnections in all charts that expect data from the deleted charts. The system then copies the chart or charts from the other project.
4. Close all open interconnections with the menu command "Options > Make Textual interconnections".  
Result: the connections are closed again both in the charts edited in the other project and in the original project in which textual interconnections arose as a result of deleting charts.

### 3.12.1 Textual Interconnections When Branching and Merging Project Data

In conjunction with branching projects and creating separate configurable units that can then be merged later again, there are certain aspects to take into account with textual interconnections.

#### Sequence and Procedure

When you distribute a project (known as the master project here), you copy part of the chart folder to one or more other chart folders, for example to temporary projects (branch projects) for different configuration engineers. Generally, this involves one or more (but not all) charts that are transferred to a different project. The interconnections to other charts are converted to textual interconnections at the inputs and deleted at the outputs.

When you return the edited charts back to the master project later (copy or move), charts with the same name are detected in the master project and you are asked whether you want to overwrite the existing object. If you answer "no", the chart is inserted as a copy ("none" with multiple charts), if you answer "yes", the existing chart is overwritten ("All" with multiple charts). Before the chart is inserted from the branch project, it is deleted in the master project, but in contrast to normal deletion (Del) textual interconnections result.

With the menu command "Options > Make Textual Interconnections", you convert these back to real interconnections.

Instead of copying charts to other projects, you can also move them (cut and paste). The advantage of copying is, however, that you still have a functional master project until you return the edited charts.

**Notes:**

- When an interconnection is broken, neither of the interconnection partners must be renamed, otherwise the textual interconnection cannot be closed again.
- Changes to charts in the master project are discarded when charts of the same name are returned to the master project from branch projects.
- An unwanted interconnection can result in the master project if, for example, interconnections between charts are modified in the branch project and only one of the charts involved is returned to the master project.  
**Example:** In chart CFC\_A, there is an interconnection to a block in chart CFC\_B. Both charts are copied to a branch project and edited there. During editing, the interconnection between the charts is deleted. Only CFC\_A is returned to the master project. A textual interconnection results in CFC\_B of the master project, that can also be closed. Result: The interconnection deleted in the branch project reappears in the master project.
- Textual interconnections created before copying/moving are included in the target project. This might be a concrete path reference (that can be closed) or a character string (required connection that will only be configured in the target project).

# 4 Test and Commissioning

## Overview

To support you when commissioning a new project, the CFC editor includes test functions allowing you to monitor and, if necessary, modify the block I/Os on the AS.

## 4.1 General

The CFC editor has two operating modes: The Edit mode and the Test mode.

- In the Edit mode, you configure the entire software structure for a CPU offline (see also Chapter 3 "Working with the CFC Editor").
- When you change to the Test mode, you can monitor and assign parameters to the block I/Os; in other words, monitor and modify values online.

During a test, there must be a unique assignment between objects in the graphic program representation and the physical addresses on the CPU. For this reason, in the Test mode, in contrast to the Edit mode, you cannot make changes that affect the structure of the user program (for example inserting or deleting blocks, changing interconnections from inputs and outputs etc.).

## Further Test Functions

- To allow you to monitor specific block I/Os in different blocks and charts of a CPU at once, you can collect these I/Os in a dynamic display window and display the constantly updated values.
- In addition to the direct test functions, there is also a group of functions that help you to prepare for testing and commissioning, for example displaying module information, setting the time of day etc.

## Requirements

To use the test and startup functions, the user program created in the CFC editor must first be compiled free of errors and downloaded to the CPU.

## 4.2 Functions Before and During the Test

In addition to the direct functions, there are also other functions that you will find helpful for testing and commissioning. These include the following:

- General functions such as "Set Time and Date"
- Functions that support you during system diagnostics and troubleshooting, such as displaying the mode or module information.

These functions are described below.

### 4.2.1 Comparing the Time Stamp of the CPU Program

You can display the time stamp of the loaded user program with the "PLC > Compare..." menu command.

Three time stamps are displayed in the dialog:

- The last download-relevant change
- Last offline program change
- Last online program change

---

**Note:**

The "Last download-relevant change" time stamp is also changed by assigning parameters to block I/Os in the Test mode.

The user program that you downloaded to the CPU is only identical with the user program in CFC and the machine code when all the time stamps are identical.

From the time stamps that match or do not match, you can see whether changes have been made in the user program and can decide whether you need to recompile or download again.

---

### 4.2.2 Starting and Stopping the CPU Program

You can start or stop the CPU program by selecting the "PLC > Operating Mode..." menu command and then clicking the appropriate button "Cold Restart"/"Hot Restart"/"Warm Restart" or "Stop" in the dialog box. CPU program:stopping

This function is identical to the corresponding function of the SIMATIC Manager. This procedure is described in the online help of the SIMATIC Manager.

### 4.2.3 Clearing/Resetting a CPU

Clearing or resetting means that the entire user program on the CPU and all the existing connections are deleted. The CPU must be in the STOP mode.

This function is identical to the corresponding function in the SIMATIC Manager described in the online help of the SIMATIC Manager.

---

**Note:**

If you download the **entire** user program, all the blocks on the CPU are deleted after you confirm a prompt. In this case, you do not need to clear the CPU memory first. When you clear the CPU memory, not only the user data of the CPU but also the connections to the module are deleted.

---

### 4.2.4 Set Time and Date

You can set the "Date and Time" on a particular CPU by selecting the "PLC > Set Time and Date..." menu command.

This function is identical to the corresponding function in the SIMATIC Manager described in the online help of the SIMATIC Manager.

### 4.2.5 Displaying Module Information

You can display the current status of the CPU module with the "PLC > Module Information..." menu command.

This function is identical to the corresponding function in the SIMATIC Manager described in the online help of the SIMATIC Manager.

## 4.3 Working in the Test Mode

To support you during commissioning, the CFC editor provides test functions that allow you to monitor and influence the execution of the blocks on the CPU and, if required, to change settings.

The Test mode relates to the CPU belonging to the currently active chart.

### Changing the Watch Cycle

I/Os that are watched in the Test mode, are updated cyclically with the values from the CPU.

The default for the update cycle time of these I/Os is two seconds. This setting is CPU-specific; in other words, it applies to all charts of the current chart folder.

You can change the cycle time in the Edit or Test mode, as follows: "Debug > Test Settings..." opens a dialog box in which you can set the time for the watching cycle.

### Test Modes

Testing can be done in one of two modes:

- Process Mode
- Laboratory Mode

You can select the mode for the test in the Edit mode using the menu commands in the "Debug" menu. It is not possible to change over once you are in the Test mode.

In the **process mode**, the communication for online dynamic display of the blocks is restricted and causes only limited extra load on the CP and bus. When the Test mode is activated, all blocks have the status "watch off".

The **laboratory mode** allows convenient and efficient testing and commissioning. In the laboratory mode, in contrast to the process mode, communication for online dynamic display of CFC charts is unrestricted. When the Test mode is activated, all blocks have the status "watch on". You can monitor the I/Os registered for testing.

### Activating the Test Mode:

Select the menu command "Debug > Test Mode" or Click the  icon in the toolbar. The Test mode is activated. You can now activate the debug menu functions; Most of the functions of the Edit mode become inactive.

If the user program has been modified in the meantime, a message will be displayed when you change to the Test mode.

### **Deactivating the Test Mode**

When you deactivate the Test mode, the test functions become inactive and the functions of the Edit mode are activated again.

You deactivate the test mode by clicking on the  icon again in the toolbar or by selecting the "Debug > Test Mode" menu command.

## 4.4 Monitoring and Assigning Parameters to Block I/Os

I/Os that are registered for testing, are supplied with the current values from the CPU in the Test mode if "watch" is activated for these I/Os.

When you activate the Test mode, the "watch on"  function is also activated for CFC charts in the "laboratory mode".

This means that in the Test mode, you can display the development of the values of block /chart I/Os registered for display; in other words, the values are read out cyclically from the CPU and displayed. You can modify the options for this dynamic display and the parameters of the I/Os in the Test mode.

In the "process mode", the "watch off" function  is activated. This means that you must first select the blocks for dynamic display (in the chart) that you want to monitor; following this, you select "watch on".

If, on the other hand, there is an overload, you can remove individual blocks/charts from the watch list (select the block/chart and then the "watch off" menu command).

**Notice:** If watching is deactivated, and you select an I/O to register it for testing with , watching will be activated for this and for all other I/Os of this block previously registered for testing.

When you activate the Test mode, the connections to the CPU are established for all I/Os listed in a dynamic display window. You activate the "watch" function with a check mark in the "Watch" column for the individual I/Os.

---

### Note:

- You cannot watch block I/Os that are not stored in DBs. This also applies, for example, to unconnected inputs of FCs or BOPs and for outputs of the data type STRING, DATE\_AND\_TIME, ANY.
  - In the Test mode, function blocks (FBs) do not indicate the value of the connected source at interconnected inputs if EN=0 is set. The value with which the block was last executed is displayed at these interconnected inputs. Remember 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 interconnected source.
-

## 4.4.1 Block and Chart I/Os in the Chart Window

### Adding/Removing I/Os in the Watch List

In the Edit mode or in the Test mode (process or laboratory mode), you can register individual block or chart I/Os for testing by selecting the I/O and then the "Debug > Inputs/Outputs > Add to Watch List" menu command or by clicking the



in the toolbar.

In the Test mode, watching is also activated at the same time; in other words, the I/O is shown on a yellow background and is displayed with its current value. If watching had been deactivated for this block/chart previously, the watch function is activated for all other I/Os that were previously added to the watch list.

You can remove I/Os from the watch list by selecting the I/O and then selecting the "Test > Remove from Watch List" menu command or by clicking the



in the toolbar.

**As an alternative:** You can also select and deselect I/Os of a block/chart for testing in the Edit mode by displaying the "Object Properties" for a selected block and selecting or deselecting the I/Os in the "Watched" column in the "Inputs/Outputs" tab.

---

#### Note:

You should avoid watching too many I/Os at one time, otherwise there is a large communication load on the bus and on the CPU. If the load is too great (for example >500 in the 1s watch cycle), the watchdog may respond; The I/O values are then marked as temporarily "disturbed".

---

### Activating and Deactivating I/O Watching

You activate watching (display of the current values of an I/O) of the block/chart I/Os in the watch list as follows:

- Automatically by activating the Laboratory Test mode.
- with the menu command "Debug > Watch On" or with the  icon in the toolbar. In the laboratory mode, this function applies to all blocks; in the process mode only to the blocks previously selected in the chart.
- By selecting the menu command "Debug > Watch Off" or by clicking the  icon, you can stop the watch; in other words, the values of the I/Os are no longer refreshed.  
In the laboratory mode, this applies to all blocks; in the process mode only to the blocks selected in the chart.

All the input and output values activated in the watch list are updated within the monitoring cycle.

### Representation of the Values

The values are displayed beside the I/Os according to their data type. They are shown on the screen on a colored background.

- **Asterisks black on yellow** = values when changing to dynamic display
- **Value black on yellow** = values read from the CPU
- **#### on a red background** = while the dynamic values from the CPU cannot be supplied (problem)

### Setting Parameters for I/Os

While the Test mode is active, you can modify the values of all unconnected inputs. Before the modified value is adopted on the CPU, its validity is checked.

Please note that parameters modified in the Test mode are always included in the CFC data management.

## 4.5 The Dynamic Display

In the Test mode, the values of block and chart I/Os can be displayed dynamically in a separate window. This is possible for the elementary data types (BO, W, R, ...) and for elements of structures.

### The Dynamic Display Window

The dynamic display window can be opened and arranged in the window of the CFC editor along with any chart windows. You can adjust the size of the window. There is only one dynamic display window without a separate menu for all the charts of a CPU. When the window is closed (or when the CFC editor is exited) the content is saved (only the static content, not the values) and loaded again when you open the window.

The dynamic display window is activated and deactivated with "View > Dynamic Display". This is divided into columns (you can vary the width) and includes the following titles and information:

- **Status:**  
In this column, a status message is displayed (for example CPU fault, value error, ...).
- **Watch**  
In a check box, you can decide whether or not the I/O is watched; in other words, whether or not the value is displayed dynamically.
- **Chart** displays the name of the chart. If the chart is a nested chart, the names of the higher-level charts (parents) are also shown (for example, Topchart\Sim\_reg ).
- **Block** Shows the name of the CFC block or nested chart (for example, switch, controller, ...)
- **I/O** Shows the name of the I/O (for example LMN, IN1, ...)
- **Value** shows the actual value of the block I/O read from the CPU (if the check box is set in the "Watch" column). If there are problems with a value ##### is displayed.
- **Unit** Shows the text for the selected physical unit (if it exists) (for example s, %, ...)
- **Comment** shows the comment for the I/O (if it exists).

**Tip:** If the text cannot be displayed completely because the column is not wide enough, you can see the complete text in a screen tip by positioning the mouse pointer on the column entry.

## 4.5.1 I/Os in the Dynamic Display Window

### Inserting I/Os

You can insert I/Os into the dynamic display in the Edit or in the Test mode. The way in which this function works is analogous to adding and removing block/chart I/Os in the chart window.

You can include I/Os from an open chart in the dynamic display window as follows:

- Select the required block I/O and then select "Debug > Inputs/Outputs > Include in Dynamic Display".
- Drag the required chart I/O to the dynamic display window with the mouse.

Block I/Os with data type **STRUCT**: Before the I/O is entered in the dynamic display, a dialog box is displayed in which you can select the structure element for the value to be displayed.

If you want to include **all I/Os** of a block or a nested chart in the dynamic display, you can select the block or chart and drag it to the dynamic display window.

Exception: I/Os of the STRUCT data type are not included.

### Deleting I/Os

Select the relevant line or lines in the dynamic display and remove it (or them) with "Edit > Delete" or the DEL key.

### Watching Values

You can watch or monitor the values of block or chart I/Os without opening the relevant charts.

The value is displayed dynamically (on a yellow background) according to the selected watch cycle when:

- The Test mode is activated,
- The I/O (or element of a structure) has been enabled for watching (selection in the "Watch" column)
- The connection to the CPU is established.

If an error is affecting a value, this is displayed as ##### and the value of an I/O that cannot be monitored is displayed as -----.

### Opening a Chart

If you select a row in the dynamic display, you can jump to the chart in which the watched I/O is located. With the menu command "Edit > Go To > Chart", the corresponding chart is opened and the I/O is selected.

# 5 Documentation

## Overview

This chapter provides you with an overview of how to create comprehensive documentation for your CFC configuration structure. This includes the following:

- The chart
- The chart reference data
- The logs

## 5.1 Printing a Chart

With the print function, you can print the current chart in CFC. You can set the layout you require ("Chart > Page Setup...") and display a preview ("Chart > Print Preview") of the chart as it will be printed on the printer.

### Printing the Active Chart

With the "Chart > Print" menu command, you print the active chart on the default printer.

A dialog box is opened in which you can select what you want to print, for example "current sheet:current chart partition:all chart partitions".

If the chart has chart I/Os, you can decide whether or not the chart is printed with or without chart I/Os. The chart I/Os are printed on a separate page in a table (containing all the relevant information such as data type, initial value, invisible, ...). If there are too many I/Os to fit on one page, further pages are printed.

When you print from the SIMATIC Manager, the lower-level charts, in other words the nested charts of the basic chart are also printed (default). When you reset the check box **Include nested charts (from SIMATIC Manager)**, you can prevent these charts being printed out.

With the "Options" button in this dialog box, you can display further print options. Regardless of the current layout in the chart (can be set with "Options > Customize > Layout..."), in this dialog, you can specify how the addresses, I/Os, and information from the block headers will appear in the printout.

## 5.1.1 Footers

### Printing with DOCPRO

With the DOCPRO optional package, you can print the CFC chart (or the CFC charts) along with footer data. In the footer data, a distinction is made between global data and local data specific to a chart.

You can enter the global data for the project using DOCPRO or the SIMATIC Manager; the local chart-specific data must be entered with the CFC editor. Remember that the data specific to a chart overwrite the entries of the global data for the particular chart.

You can also enter data specific to a chart even when you have not installed the DOCPRO optional package. This data is saved and can be printed later when DOCPRO is available for print jobs.

The **footer** includes a fixed field with the continuous page number of the print job. With jobs 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 however the page number is followed by a "+" to indicate that there are more pages to come; for example 1+ .... 7+ etc.

The number of the last page is followed by a "-"; for example 127-.

In the global footers, you can enter **keywords** that are replaced by the actual texts when you print out. The current texts are taken from the Object Properties of the chart (for example chart comment, date created etc.). The keywords you can use and their meaning (origin) are explained in the online help for CFC.

### Footer Data for a Specific Chart

With the "Chart > Footers..." menu command, you display a dialog box with several tabs: "Part 1" to "Part 4" and "Free Fields".

Here, you can make entries in the active fields, such as:

Type of document, Date created, Document number, Date modified, Free texts etc.

## 5.2 Chart Reference Data

### Overview

With the menu command "Options > Chart Reference Data..." or by clicking the  icon in the toolbar, you can start and application and display and print out various information in lists or as a tree structure (see 5.2.1).

These lists help you to check your configuration structure.

### Various Versions

A list created once by the system is not updated automatically. Each list provides information about the status of the project at the time the list was created. Several versions of these lists can be opened at the same time.

If required, you can update the list using the "View > Update" menu command or with the "F5" key.

### Further Functions

You can generate lists not only of the project currently being edited in CFC, but also access all other CFC projects. This allows you to compare several projects. Lists of several projects can be opened at the same time.

With the "Edit > Find..." menu command, you can search for terms (text strings) in the active window. If a term is found, the location is displayed.

If you double-click the name in the first column, you can open the chart in which the point of use is located. The relevant object is then selected.

### Documentation

Along with the printed charts, the reference data provide you with complete documentation of your configuration structure.

### Export

You can save the generated list as a file in the CSV format (for example for use in EXCEL) with the "Reference Data > Create Export File..." menu command.

## 5.2.1 Lists of the Chart Reference Data

You can display and print the following chart reference data:

- **Run Sequence**  
Graphic display of the entire run sequence of a CPU.
- **Cross-References Chart Element -> Address**  
The list includes all the global addresses used in the project with the elements that access them.
- **Cross-References SFC -> Chart Element**  
The list includes the accesses by all SFC charts to the I/Os of CFC chart elements.
- **Cross-References Chart Element -> Run-Time Group**  
The list includes the accesses to all run-time groups by all CFC and SFC charts.
- **Block Types**  
The list includes block types used and their locations (CFC chart).
- **S7 Resource Allocation**  
The list includes the assignment between CFC configuration objects and S7 resources.
- **Local Data**  
This list includes all the OBs in the program along with the calculated local data requirements and the local data size of the individual priority classes configured offline and actually existing online.
- **Block Call Hierarchy**  
Displays a graphic representation of the call hierarchy of all blocks in the current program.
- **Textual Interconnections**  
This list contains all textual interconnections (path reference to the interconnection target) with the chart name and chart element of the interconnection source.
- **Statistics**  
Graphic display of the numbers of all CFC and SFC objects and S7 resources used as well as the time stamp of the current program and the process objects of the project.

For a detailed description of the lists and the meanings of the columns, refer to the online help of CFC.

## 5.3 Logs

The logs do not belong to the documentation of the configuration structure but can provide important information during configuration and trial startups if errors occur.

For further information on the logs, refer to the online help of the "Logs" dialog box.

# A Technical Specifications

## A.1 [S7] Technical Specifications

### Hardware Requirements

- SIMATIC PG or PC
- Pentium
- RAM at least 128 Mbytes  
(recommended 256 Mbytes or more)
- Hard disk (free space) approx. 500 Mbytes (less RAM memory)
- Graphics card VGA 640 x 480  
(recommended: SVGA 1024 x 768 or higher)
- MPI connection for online operation
- SIMATIC S7-300, S7-400

### Software Requirements

- Microsoft Windows 95 (with Service Pack 1)  
or
- Microsoft Windows 98  
or
- Microsoft Windows NT (with Service Pack 3)  
or
- Microsoft Windows ME  
or
- Microsoft Windows 2000  
or
- Microsoft Windows XP
- STEP 7
- SCL Compiler

## A.2 Field/Name Lengths and Conventions

Object	Length	Remarks
Chart	22	Must not include the following characters: \ . " %
Chart comment	255	All ANSI characters are permitted.
Run-time group	22	Character set as for "chart"
Task	22	Character set as for "chart"
Block type	8	Determined by STEP 7.
Comment for block instance	80	All ANSI characters are permitted.
Parameter name	24	Determined by STEP 7. The 1st character must not be a number (0 - 9).
Parameter comment	80	All ANSI characters are permitted.
Name of the block instance	16	Must not include the following characters: \ . " %
Global variables	24	Characters determined by the symbol editor; all permitted even blanks and special characters except ".

### A.3 [S7] Data types

Abb.	Keyword	Meaning	Bits
A	ANY	Pointer to data element	80
BO	BOOL	Logical number	1
BY	BYTE	Sequence of 8 bits	8
C	CHAR	Single character	8
CR	COUNTER	Number of an S7 counter	16
D	DATE	Date	16
DB	BLOCK_DB	Number of a DB	16
DI	DINT	Double integer	32
DT	DATE_AND_TIME or DT	Date and time	64
DW	DWORD	Sequence of 32 bits	32
FB	BLOCK_FB	Number of an FB	16
FC	BLOCK_FC	Number of an FC	16
I	INT	Integer	16
P	POINTER	Pointer to memory area	48
R	REAL	Floating-point number	32
S	STRING	Text string of any length	256 bytes
SD	BLOCK_SDB	Number of an SDB	16
SN	STRING[n]	Text string with maximum n characters, $1 \leq n \leq 253$	
ST	STRUCT	Parenthesis for elementary data types and other structures (nesting depth: 8)	
T	TIME_OF_DAY or TOD	Timer	32
TI	TIME	Duration	32
TR	TIMER	Number of an S7 timer	16
T5	S5TIME	Duration in S5 format	16
W	WORD	Sequence of 16 bits	16

For a detailed description of the data types, refer to the online help.



## **B Abbreviations**

<b>BOP</b>	Basic Operation
<b>C / C++</b>	High-level language for programming computers
<b>CFC</b>	Continuous Function Chart
<b>CPU</b>	Central Processing Unit
<b>DB</b>	Data Block
<b>ES</b>	Engineering System
<b>FB</b>	Function Block
<b>FC</b>	Function Code
<b>HID</b>	Higher Level Designation
<b>IEA</b>	Import/Export Assistant
<b>LAD</b>	Ladder Diagram
<b>OB</b>	Organization Block
<b>OCM</b>	Operator Control and Monitoring (WinCC)
<b>OS</b>	Operator Station

<b>PC</b>	Personal Computer
<b>PCS 7</b>	Process Control System (SIMATIC)
<b>PG</b>	Programming Device
<b>PH</b>	Plant Hierarchy
<b>PLC</b>	Programmable (Logic) Controller
<b>SFB</b>	System Function Block
<b>SFC</b>	Sequential Function Chart
<b>SINEC</b>	Siemens Network Architecture
<b>STEP 7</b>	Software development environment for SIMATIC S7 / M7
<b>STL</b>	Statement List

# Glossary

## A

In S7 the tasks are implemented as → organization blocks (OBs).

### **Absolute addressing**

If data is addressed in absolute form, the absolute address is used to access the value with which the operation will be performed. Example: The address Q4.0 identifies bit 0 in byte 4 of the process output image (PIQ).

### **Access**

SFC elements or block I/Os can access addresses, charts, block I/Os, or run-time groups. A distinction is made between read and write access.

### **Address**

An address is the identifier of the location of data or an area containing data, examples: input I12.1; memory word MW25; data block DB3. An address is part of a STEP 7 instruction and tells the processor what it should perform an operation on. Addresses can be specified in absolute or symbolic form.

In SFC, the address is part of a statement (step) or condition (transition).

## B

### **Basic operation**

Basic operations (BOPs) are permanent objects in the system that perform simple functions such as AND, OR etc. They are represented in CFC as blocks.

### **Block:**

Blocks are separate parts of a user program that are distinguished by their function, their structure or purpose.

CFC works with ready-made block types that are placed (inserted) in a CFC chart. When a block type is inserted in the chart, a block instance is created. These block instances and their graphic representation are blocks in the sense of CFC.

By positioning them on a free location in the sheet, they become normal blocks again including their previous interconnections.

### **Block category**

The block category identifies different forms of blocks. Block categories include, for example, data blocks (DBs), function blocks (FBs), and functions (FCs).

### **Block I/O**

Block input or block output

### **Block input**

Block I/O that can be interconnected with block outputs and → addresses of the same data type or can have parameters assigned to it.

### **Block instance**

A block instance is the implementation of a block type. A block type inserted in a CFC chart becomes a block instance. When it is inserted, the block instance is assigned run-time properties and is given a name that is unique within the chart.

### **Block header**

The top part of a block when displayed in CFC. It contains the name and the task assignment (run-time property).

### **Block icon**

Graphic representation of the most important information of a technological PLC block that can be controlled and monitored by the operator. The block icon is normally placed in an OS overview picture. The corresponding faceplate can be called via the block icon.

### **Block library**

→ Library

### **Block output**

Block I/O that can be interconnected with block inputs and → addresses of the same data type.

### **Block type**

Block types are ready-made program sections that can be used in a CFC chart (for example, controllers, multiplexers etc.). When they are inserted, block instances are created. You can create any number of block instances from a block type.

The block type determines the characteristics (algorithm) for all implementations of this type. The name of the block is entered in the symbol table.

## C

### Catalog

This is a window in CFC that can be switched over between the catalogs of the **blocks**, **charts**, **libraries** and **unplaced blocks** (if unplaced blocks exist). Blocks, charts, and text elements can be inserted in the chart from the catalog.

### CFC

Continuous Function Chart.

1. Function chart (CFC chart) with the graphic interconnection of technological functions (blocks).
2. An editor for plant-oriented, graphic configuration of automation tasks. Using CFC, entire software structures are created (CFC charts) from ready-made blocks.

### Chart

In the ES, a chart is one of the following depending on its context:

- A CFC chart consisting of 1 to 26 chart partitions each with six sheets and possibly overflow pages.
- A nested chart results when a chart is inserted in another chart (chart-in-chart).
- An SFC chart consisting of one to a maximum of eight tabs. Each tab contains a sequence.

### Chart folder

Folder in the project structure containing charts of a user program.

### Chart installation pointer

The installation pointer decides the next insert point for installation in the run sequence. There is a chart installation pointer (installs run-time groups for a chart) and a block installation pointer (installs blocks).

### Chart overview

The overview of a CFC chart or chart partition with its six sheets.

### Chart partition

Part of a CFC chart. A CFC chart is subdivided into a maximum of 26 chart partitions (named A to Z) each with 6 sheets.

### **Component view**

Device-oriented view in the SIMATIC Manager. The project is displayed with its components (station, module, program ... ); alternative to the → Plant View.

### **Connector**

The connection point on the block with a reference to the I/O to which it is connected. Connectors are used when no more connecting lines can be drawn on a sheet because it is already full. The connectors allow complex CFC structures, including those extending to other sheets, to be displayed completely.

### **Consistency Check**

Checks the consistency of block types, shared addresses etc. of the chart folder.

### **CPU**

The CPU (central processing unit) is a module in the programmable controller on which the user program is stored and run. It contains the operating system and the communications interfaces.

### **Cycle time**

The cycle time is the time required by the CPU to execute the user program once.

### **D**

### **Data block (DB)**

Data blocks are data areas in the user program containing user data. There are shared data blocks that can be accessed by all logic blocks and instance data blocks that are assigned to a particular FB call. In contrast to all other blocks, they contain no instructions.

### **Data type**

A data type specifies how the value of a variable or constant at a block I/O will be used. "BOOL", for example, defines a binary variable and "INT" defines a 16-bit fixed-point variable.

**Download-relevant change**

A modification in the CFC/SFC chart (SFC type) is download-relevant when it causes a change in the user program of the CPU (for example a change to an interconnection). Changes that are not download-relevant involve, for example, block positions and comments.

**Drag-and-drop**

Using drag-and-drop, you can move, copy, and insert objects with the mouse.

Procedure:

1. Select the relevant objects by clicking on them or drawing a lasso
2. Move the mouse pointer onto an object, click the left mouse button and hold it down
3. To copy, hold down the CTRL key - the mouse pointer has a "+" icon added
4. Move the mouse to the required position and release the mouse button, the objects are inserted.

**Dynamic display**

In the dynamic display, input and output values of a block in a CFC chart or address values in an SFC chart are updated from the CPU in the Test mode.

**Dynamic display window**

Window in CFC for monitoring selected block I/Os on CPUs.

**E****Edit mode:**

(Alternative to the → Test mode) In **CFC**, blocks can be inserted, copied, moved, deleted, assigned parameters, or interconnected.

**Enable attribute**

The enable attribute is a run-time attribute. It switches, for example, a run-time group on or off. As long as it is deactivated (FALSE), the run-time group is not executed regardless of any other conditions.

The enable attribute can be set dynamically. In this case, the output value of a CFC block or the statement in an SFC action decides whether the run-time group or SFC chart is activated or deactivated.

## ES

Acronym for "Engineering System". Configuration system with which the process control system can be created and adapted to the tasks conveniently and visually.

## External view

The external view is the graphic representation of a CFC chart as a block (with interface) allowing external interconnections of the I/Os to be made. The external view has the standard interface that is derived from the SFC run-time system.

The external view of the SFC chart can be interconnected with any objects (CFC blocks, nested charts, run-time groups, textual references, shared addresses). This allows the SFC chart to be controlled directly by CFC interconnections.

The external view is displayed in a separate window in CFC. Further objects cannot be placed in this window. Interconnections to other CFC objects are implemented only via the sheet bar.

## F

### Function (FC)

(FUNCTION) According to IEC 1131-3, functions are logic blocks without memory. A function allows parameters to be transferred in the user program. Functions are suitable for programming frequently recurring complex functions, for example calculations.

See also → Block category.

### Function block (FB)

(FUNCTION BLOCK) According to IEC 1131-3, a function block is a logic block with static data. An FB allows the transfer of parameters in the user program. Function blocks are suitable for programming frequently recurring complex functions, for example closed-loop controls. Since an FB has a memory (instance data block), its parameters (for example outputs) can be accessed at any time and any point in the user program.

See also → Block category.

## H

### Higher level designation (HID)

This is made up of the hierarchical path of the → plant hierarchy.

## I

### Icon

In its graphic representation in the header, the block can have an icon instead of a textual type name (or FB/FC assignment). This icon can, for example, be a standard icon for controller, limiter, NAND element etc.

### Import/Export Assistant (IEA)

Software component in PCS 7 for handling models and process tag types.

### Installation and commissioning

The CFC/SFC editors provide test functions to allow you to monitor, modify, and change parameter settings on the PLC during installation and commissioning.

### Interconnection

This connects a block/chart I/O with another element. The value of the interconnected input is fetched from the other end of the interconnection during run time.

### Interface

The interface consists of the inputs and outputs of a block or a chart that can be interconnected and assigned parameters.

### I/O

The input or output of a block or chart. I/Os of the same data type can be interconnected with each other or with shared addresses. Within a block, an I/O is a parameter that receives data for further processing (input) and transfers the data of the result (output). The I/O of a chart has connections to selected internal block or chart I/Os.

## L

### **Library**

A folder for objects that can be used again and again. A library is not project-related. Blocks are made available in block libraries sorted according to certain criteria (block families, alphabetical order etc.).

### **Local data**

Local data are data assigned to a logic block that are listed in its declaration section or its variable declaration.

## M

### **Master project**

A master project is a project that can be distributed in several parts to other projects (branch projects) for more efficient engineering. After editing, the parts are returned to the original project = master project.

### **Message class**

The message class determines the properties of a message. In SIMATIC PCS 7, there are the following message classes: alarm, warning, tolerance, PLC and OS process control message, process message, operator prompt, and operator message.

### **Message configuration**

Creating messages with their attributes and texts. Messages can be configured in CFC/SFC.

## N

### **Nested chart**

A CFC chart that is inserted in another CFC chart (nested or top chart). Nested charts are not displayed in the SIMATIC Manager.

## O

### **Online/Offline**

In the SIMATIC Manager, objects of the programmable controller are displayed in the online view and the objects of the ES in the offline view. Online, there is a data connection between the PLC and the programming device/PG, offline there is no connection.

### **Operating system**

General term for all the functions that control and monitor the execution of the user programs, the distribution of resources to the individual programs, and the maintenance of the operating mode in conjunction with the hardware (for example standard operating system MS-WINDOWS, real-time operating system M7 RMOS32).

### **Organization block (OB)**

Organization blocks form the interface between the operating system of the CPU and the user program. The order of execution of the user program is specified in organization blocks.

An organization block corresponds to a → task.

## OS

Operator Station. A station for controlling and monitoring the process. In PCS 7, the WinCC software system is used for the OS with which all the process monitoring and control functions can be implemented.

### **Overlapping Block**

Overlapping blocks are blocks that could not be inserted completely at a free position on the sheet. They cover other objects partly or completely. Overlapping blocks are displayed in the chart in their full size but without block I/Os and in light gray. Existing I/Os are not visible in this display form (nor are the entries in the sheet bar), they do, however, exist.

The same applies to nested charts.

### **Overflow Page**

(CFC) If there are so many entries in the → sheet that there is no more space in the sheet bars, an overflow page is automatically created. An overflow page consists only of the → sheet bars with entries and contains no other objects.

## P

### Parameter

A parameter is

1. the value of a CFC block/chart I/O.
2. a variable of an S7 logic block (actual parameter, formal parameter)

### Phase offset

The phase offset shifts the point of activation of the run-time group within a task by a defined time compared with the basic cycle. Phase offset allows a uniform distribution of load within the CPU. See also → scan rate.

### Plant hierarchy (PH)

A hierarchy structured according to technological aspects.

### Plant view

View in the SIMATIC Manager based on technological aspects (plant, unit, function ...); alternative to the → component view ↑ and process object view.

## PLC

PLC is used as a general term for the unit containing the CPUs on which the user program runs. In this case, it includes S7 programmable controllers and M7 automation computers.

### Process image

Reserved areas in the RAM of the CPU. Signal states of the input and output modules are entered here.

### Process object view

View in the SIMATIC Manager. With the process object view, all the data of the basic control throughout a project can be displayed in a process control-oriented view.

### Process tag

CFC chart configured for the basic control of a process implementing a specific process control function, for example level control. The process tag can be created by copying a process tag type and then adapted to the required specific task.

**Process variable**

The process variable is a neutral object in terms of resources. It is used to connect the PLC configuration world (STEP 7, CFC ...) with the OS configuration world (WinCC). It contains information about its location during run time (for example the network address and the memory area on the PLC) as well as information on specific OS-relevant properties.

**Program**

General term for S7 and M7 programs.

**Programmable controller**

A programmable controller (PLC) is a controller in SIMATIC S7, a compact device (PLC with integrated operator controls) SIMATIC C7, or a SIMATIC M7 automation computer.

**Programming device (PG)**

Portable and compact personal computer specially designed for use in an industrial environment. A programming device is completely equipped for programming SIMATIC programmable controllers.

**Project**

A folder for all the automation solutions regardless of the number of stations or modules and how they are networked.

**R****Reference data**

Chart reference data are data available to the user in addition to the graphic chart display in the form of lists, for example the list of accesses to shared addresses or the block hierarchy calls.

**Resources**

Resources are pools of objects (FBs, FCs, DBs, OBs, bit memory, counters, timers etc.) that can be accessed when configuring and setting parameters for a CFC/SFC chart.

**Run-time attribute**

Each → run-time group has run-time attributes that control its activation. The group passes on these attributes to all the blocks it contains.

### **Run-time group**

Run-time groups are used to structure → tasks. The blocks are installed sequentially in the run-time groups. Run-time groups can be activated and deactivated separately. If a run-time group is deactivated, the blocks it contains will no longer be activated.

### **Run-time properties**

The run-time properties of a block decide how the block is included in the run sequence within the entire structure of the PLC. These properties are decisive for the response of the PLC in terms of reaction times, dead times, or the stability of time-dependent structures, for example closed loops.

## **S**

### **S7 program**

A folder for the → symbol table, the blocks, the source files, and the charts for the programmable S7 modules.

### **Scan rate**

The scan rate is a → run-time attribute. It specifies whether a → run-time group is executed every time a task is executed or only every nth time. See also → phase offset.

### **SCL**

A high-level language similar to Pascal complying with IEC 1131-3 for programming complex solutions on a PLC, for example algorithms and data processing jobs.

### **Sequential control system**

A sequential control system switches from one step to the next dependent on conditions. In PCS 7, sequential control systems are implemented with SFC charts/types.

### **SFC**

An SFC chart represents a → sequential control system that runs as a separate control system within the programmable controller.

**Shared address**

Shared addresses are objects that can be accessed by every logic block (FC, FB, OB). They include bit memory (M), inputs (I), outputs (Q), timers (T), counters (C), and elements of data blocks (DB). To access shared addresses, you can use absolute addresses or symbols.

**Shared data block**

This is a data block that can be accessed by all blocks in the program. Each CFC block instance can therefore read the shared data from the block or write data to the block.

**Sheet**

Subdivision of a chart partition. A → chart partition consists of 6 sheets. The sheet represents a working area (with sheet bars), on which blocks can be placed, assigned parameters, and interconnected.

**Sheet bar**

Margin at the right and left edges of a CFC chart. The sheet bars contain:

- references to connected objects (block/chart interface, address, and run-time group) that are not located on the active sheet
- the number of the connector reference when the connecting line cannot be drawn to the sheet bar because the chart is full.

**SIMATIC Manager**

Graphic user interface for SIMATIC users under Windows 95/98/NT/2000/XP. The SIMATIC Manager is used, for example, to create projects and access libraries.

**Structure**

A structure is a structured → data type made up of various elements. An element itself can be an elementary or a structured data type.

**Symbol**

A symbol is a name defined by the user according to certain syntactical rules. After its purpose has been defined (for example to represent a variable, data type, jump label, block), it can be used in programming and in operator monitoring and control.

Example: address : I5.0, data type: BOOL, symbol: Emer stop.

## Symbol table

Table for assigning symbols (= name) to addresses for shared data and blocks.

Examples:

<b>Symbol</b>	<b>Address</b>
Emer stop	I1.7,
Control	FB24

## Symbolic addressing

If data is addressed using symbols, the → address to be processed is specified as a symbol (not as an absolute address). Symbols are assigned to addresses in the symbol table.

## T

### Task

Tasks form the interface between the operating system of the CPU and the user program. The order of execution of the user program is specified in tasks. In M7, the tasks are mapped on RMOS tasks

A task corresponds to an organization block (OB) in S7.

### Test mode

(Alternative to the → Edit mode) Mode of CFC/SFC for testing and optimizing user program running online on the CPU.

### Top chart

A CFC chart that is not nested in another chart and can be displayed in the SIMATIC Manager ( → nested charts).

## U

### Update cycle

In the Test mode, this specifies the intervals at which the watched block I/Os are updated.

## V

### Value identifier

Symbolic substitute (text) for defined values of block I/Os of the data types BOOL, BYTE, INT, DINT, WORD and DWORD.

# Index

## A

Adding/removing I/Os in the watch list ..... 4-7

## B

Basic operation ..... 1-8  
Block comment ..... 3-17  
Block family ..... 1-7  
Block I/O  
  setting values ..... 4-8  
Block icon ..... 3-17  
Block input  
  inverted ..... 3-19  
Block installation pointer ..... 3-32  
Block instance ..... 1-6  
Block libraries ..... 1-9  
Block type ..... 1-6  
Block type change  
  central ..... 3-11  
Blocks  
  aligning ..... 3-15  
  copying ..... 3-16  
  deleting ..... 3-16  
  importing ..... 3-9  
  moving ..... 3-16  
  new version ..... 3-10  
  overlapping ..... 1-8  
BOP ..... 1-8

## C

Catalog ..... 1-9  
Changes  
  compiling ..... 3-42  
  Changing the watch cycle ..... 4-4  
Chart ..... 1-3  
  copying ..... 3-3  
  creating ..... 3-2  
  open ..... 3-2  
Chart I/Os ..... 3-5  
Chart installation pointer ..... 3-32, 3-35  
Chart partition ..... 1-3  
Chart partitions  
  inserting/deleting ..... 3-4  
Chart reference data ..... 5-3  
Chart-in-chart ..... 3-7  
Closing textual interconnections ..... 3-43  
Commissioning  
  preparing ..... 4-2

Compile  
  block type ..... 3-45  
  program ..... 3-42  
Compiling ..... 3-42  
  changes ..... 3-42  
Consistency check ..... 3-43  
Copying  
  chart ..... 3-3  
  to a different CPU ..... 3-16  
CPU program  
  starting ..... 4-2  
  stopping ..... 4-2

## D

Data type extension ..... 3-12  
DB ..... 3-42  
Delete Empty Run-Time Groups ..... 3-43  
Displaying module information ..... 4-3  
DOCPRO ..... 5-2  
Download  
  program ..... 3-47  
Downloading F programs ..... 3-47  
Driver and Message Concept ..... 3-38  
Dynamic display ..... 4-9

## E

Edit mode ..... 4-1  
Enable attribute ..... 3-22

## F

FB ..... 1-8  
FC ..... 1-8, 3-42  
Footers ..... 5-2

## G

Generate module drivers ..... 3-38, 3-43  
Generic block ..... 1-7

## H

H CPU  
  downloading ..... 3-47  
  read back ..... 3-49

- I**
- Importing blocks types ..... 3-9
  - Include user data blocks ..... 3-48
  - Interconnection
    - copying ..... 3-27
    - delete ..... 3-28
    - rewiring ..... 3-28
    - to chart I/Os ..... 3-5
    - to run-time group ..... 3-22
    - to SFC charts ..... 3-26
    - to shared addresses ..... 3-22
- K**
- Keywords (DOCPRO) ..... 5-2
- L**
- Laboratory mode ..... 4-4
  - Libraries ..... 1-9
  - Logs ..... 5-4
- M**
- Message block ..... 1-10
  - Metaknowledge ..... 3-39
  - Multiple instance block ..... 1-7
  - Multiple users ..... 3-1
  - Multi-user mode ..... 3-1
- N**
- Nested chart ..... 1-3, 3-7
  - Number conflict ..... 3-10
- O**
- Object properties ..... 3-17
  - Operator control and monitoring ..... 1-10
  - Operator control block ..... 1-10
  - Optimize code ..... 3-45
  - Optimize run sequence ..... 3-37
  - Organization block ..... 1-7
  - Overflow page ..... 1-3
  - Overlapping blocks ..... 1-8
  - Overview ..... 1-4
- P**
- PLC-OS communication ..... 1-10
  - Printing a chart ..... 5-1
  - Process mode ..... 4-4
- R**
- Read back ..... 3-49
  - Resetting a CPU ..... 4-3
  - Resources on the CPU ..... 3-42
  - Run-time group ..... 3-2, 3-35
    - creating ..... 3-36
    - interconnecting ..... 3-22
  - Run-time properties ..... 3-30
- S**
- Set time and date ..... 4-3
  - Several configuration engineers ..... 3-50
  - SFC access ..... 3-26
  - Shared address ..... 3-22
  - Sheet ..... 1-3
  - Sheet bar jump ..... 3-27
  - Sheet view ..... 1-5
  - Signal tracking ..... 3-27
  - SIMATIC Manager ..... 1-2
  - STEP 7 environment ..... 1-2
  - Structure ..... 3-29
  - System attributes
    - for chart I/Os ..... 3-5
- T**
- Test
    - preparing ..... 4-2
  - Test mode ..... 4-1, 4-4
  - Test modes ..... 4-4
  - Text box ..... 3-15
  - Textual interconnection ..... 3-23
  - Textual interconnections ..... 3-3
  - Time stamp ..... 4-2
  - Tolerant type import ..... 3-12
  - Top chart ..... 3-7
- U**
- Update Sampling Time ..... 3-43
  - User program
    - downloading ..... 3-46
- V**
- Value identifier ..... 3-20
- W**
- WinCC ..... 3-13
- X**
- XML files ..... 3-39