SIEMENS

SINAMICS/SIMOTION

Motion Control
SINAMICS/SIMOTION DCC Editor Description

Programming and Operating Manual

Legal information

Warning notice system

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

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

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

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

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

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

Qualified Personnel

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

Proper use of Siemens products

Note the following:

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

Trademarks

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

Disclaimer of Liability

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

Preface

SIMOTION Documentation

An overview of the SIMOTION documentation can be found in the SIMOTION Documentation Overview document.

This documentation is included as electronic documentation in the scope of delivery of SIMOTION SCOUT. It comprises ten documentation packages.

The following documentation packages are available for SIMOTION V4.4:

- SIMOTION Engineering System Handling
- SIMOTION System and Function Descriptions
- SIMOTION Service and Diagnostics
- SIMOTION IT
- SIMOTION Programming
- SIMOTION Programming - References
- SIMOTION C
- SIMOTION P
- SIMOTION D
- SIMOTION Supplementary Documentation

SINAMICS documentation

The SINAMICS documentation is organized into 2 parts:

- General documentation/catalogs
- Manufacturer/service documentation

A current overview of the documentation in the available languages can be found on the Internet:

http://www.siemens.com/motioncontrol

Select the menu items "Support" --> "Technical Documentation" --> "Overview of Publications."

The Internet version of DOConCD (DOConWEB) is available on the Internet:

http://www.automation.siemens.com/doconweb

Information on the range of training courses and FAQs (Frequently Asked Questions) are available on the Internet:

http://www.siemens.com/motioncontrol

Follow the menu item "Support".

Further documentation for the DCC editor

- SINAMICS/SIMOTION Function Manual, *Description of the standard DCC blocks*

Hotline and Internet addresses

Additional information

Click the following link to find information on the the following topics:

- Ordering documentation / overview of documentation

- Additional links to download documents

- Using documentation online (find and search manuals/information)

http://www.siemens.com/motioncontrol/docu

Please send any questions about the technical documentation (e.g. suggestions for improvement, corrections) to the following e-mail address:
docu.motioncontrol@siemens.com

My Documentation Manager

Click the following link for information on how to compile documentation individually on the basis of Siemens content and how to adapt it for the purpose of your own machine documentation:

http://www.siemens.com/mdm

Training

Click the following link for information on SITRAIN - Siemens training courses for automation products, systems and solutions:

http://www.siemens.com/sitrain

FAQs

Frequently Asked Questions can be found in SIMOTION Utilities & Applications, which are included in the scope of delivery of SIMOTION SCOUT, and in the Service&Support pages in **Product Support**:

http://support.automation.siemens.com

Technical support

Country-specific telephone numbers for technical support are provided on the Internet under **Contact**:

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

Table of contents

Fundamental safety instructions

1.1 General safety instructions

⚠ WARNING
Risk of death if the safety instructions and remaining risks are not carefully observed
If the safety instructions and residual risks are not observed in the associated hardware documentation, accidents involving severe injuries or death can occur. ● Observe the safety instructions given in the hardware documentation. ● Consider the residual risks for the risk evaluation.

⚠ WARNING
Danger to life or malfunctions of the machine as a result of incorrect or changed parameterization
As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death. ● Protect the parameterization (parameter assignments) against unauthorized access. ● Respond to possible malfunctions by applying suitable measures (e.g. EMERGENCY STOP or EMERGENCY OFF).

1.2 Industrial security

Note

Industrial security

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.automation.siemens.com

---

⚠ **WARNING**

Danger as a result of unsafe operating states resulting from software manipulation

Software manipulation (e.g. by viruses, Trojan horses, malware, worms) can cause unsafe operating states to develop in your installation which can lead to death, severe injuries and/ or material damage.

- Keep the software up to date.
  Information and newsletters can be found at:
  http://support.automation.siemens.com

- Incorporate the automation and drive components into a state-of-the-art, integrated industrial security concept for the installation or machine.
  For more detailed information, go to:
  http://www.siemens.com/industrialsecurity

- Make sure that you include all installed products into the integrated industrial security concept.

Introduction

<div style="text-align: right">2</div>

**Drive Control Chart (DCC) for SINAMICS and SIMOTION means
graphic configuration and expansion of the device functionality by means of freely available
control, arithmetic and logic blocks**

Drive Control Chart (DCC) expands the facility for the simplest possible configuring of
technological functions both for the SIMOTION motion control system and the SINAMICS drive
system. This opens up a new dimension for users for adapting the specified systems to the
specific functions of their machines. DCC has no restriction with regard to the number of usable
functions; this is only limited by the performance capability of the target platform.
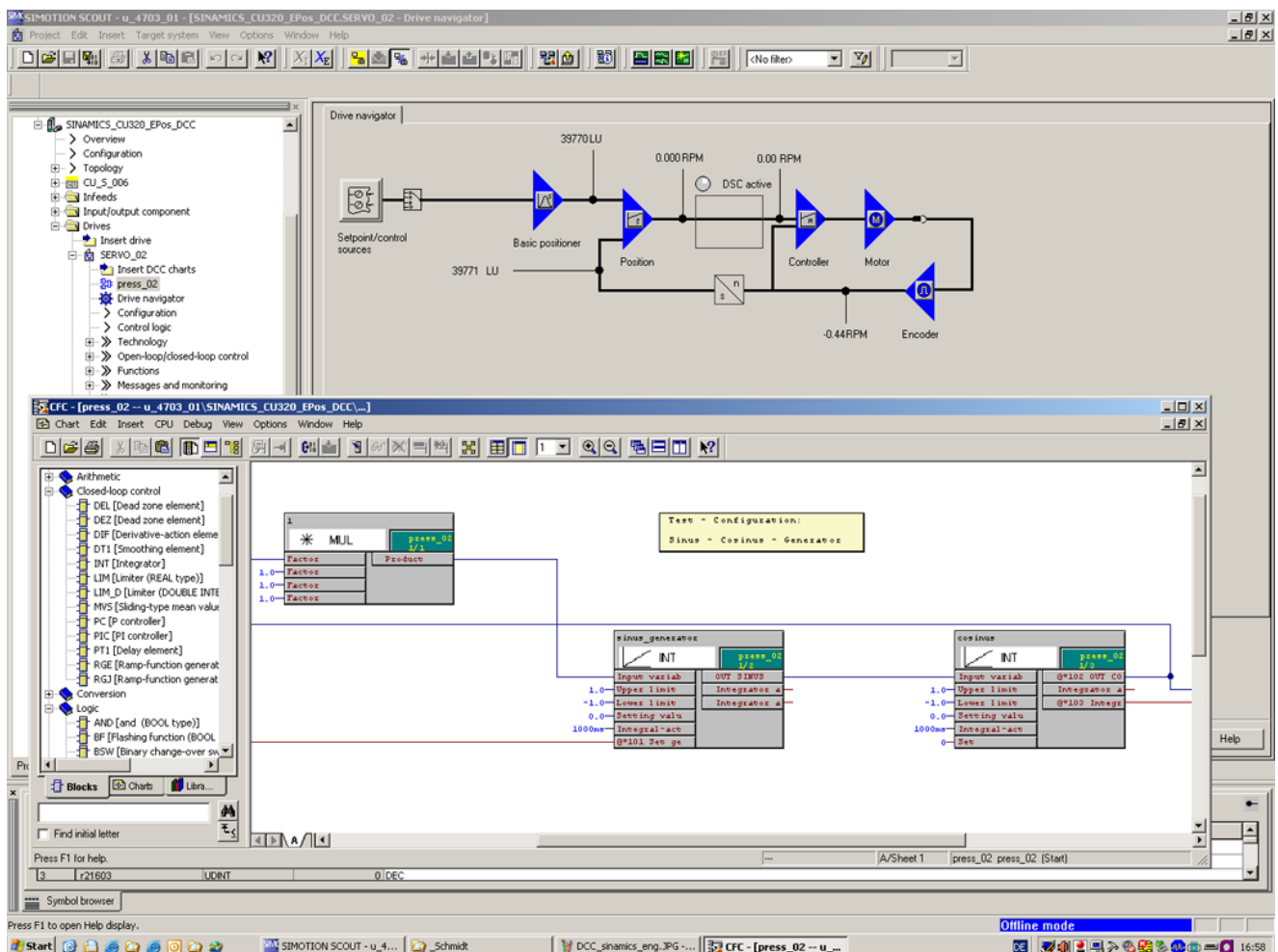
DCC comprises the DCC editor and the DCB library (block library with standard DCC blocks).

The user-friendly **DCC editor** enables easy graphic configuration and a clear representation
of control loop structures as well as a high degree of reusability of existing charts.
The open-loop and closed-loop control functionality is defined by using multi-instance-capable
blocks (**Drive Control Blocks, DCBs**) from a pre-defined library (**DCB library**) that are selected

and graphically linked by dragging and dropping. Test and diagnostic functions allow verification of program behavior or the identification of causes in the event of errors.

The block library contains a large selection of control, arithmetic and logic blocks as well as extensive open-loop and closed-loop control functions.

All commonly used logic functions are available for selection (AND, XOR, On/Off delay, RS flip-flop, counters, etc.) for the logic operation, evaluation and acquisition of binary signals. Diverse arithmetic functions such as absolute-value generation, dividers and minimum/ maximum analysis are available for monitoring and evaluating numeric variables. In addition to the drive control, axial winder functions, PI controllers, ramp-function generators or sweep generators can be configured simply and without problem.

Almost unlimited programming of control structures is possible in conjunction with the SIMOTION motion control system. These can then be combined with other program sections to form an overall program.

Drive Control Chart for SINAMICS drives also provides a convenient basis for resolving drive-level open-loop and closed-loop control tasks directly in the converter. This results in further adaptability of SINAMICS for the tasks set. On-site processing in the drive supports modular machine concepts and results in increased overall machine performance.
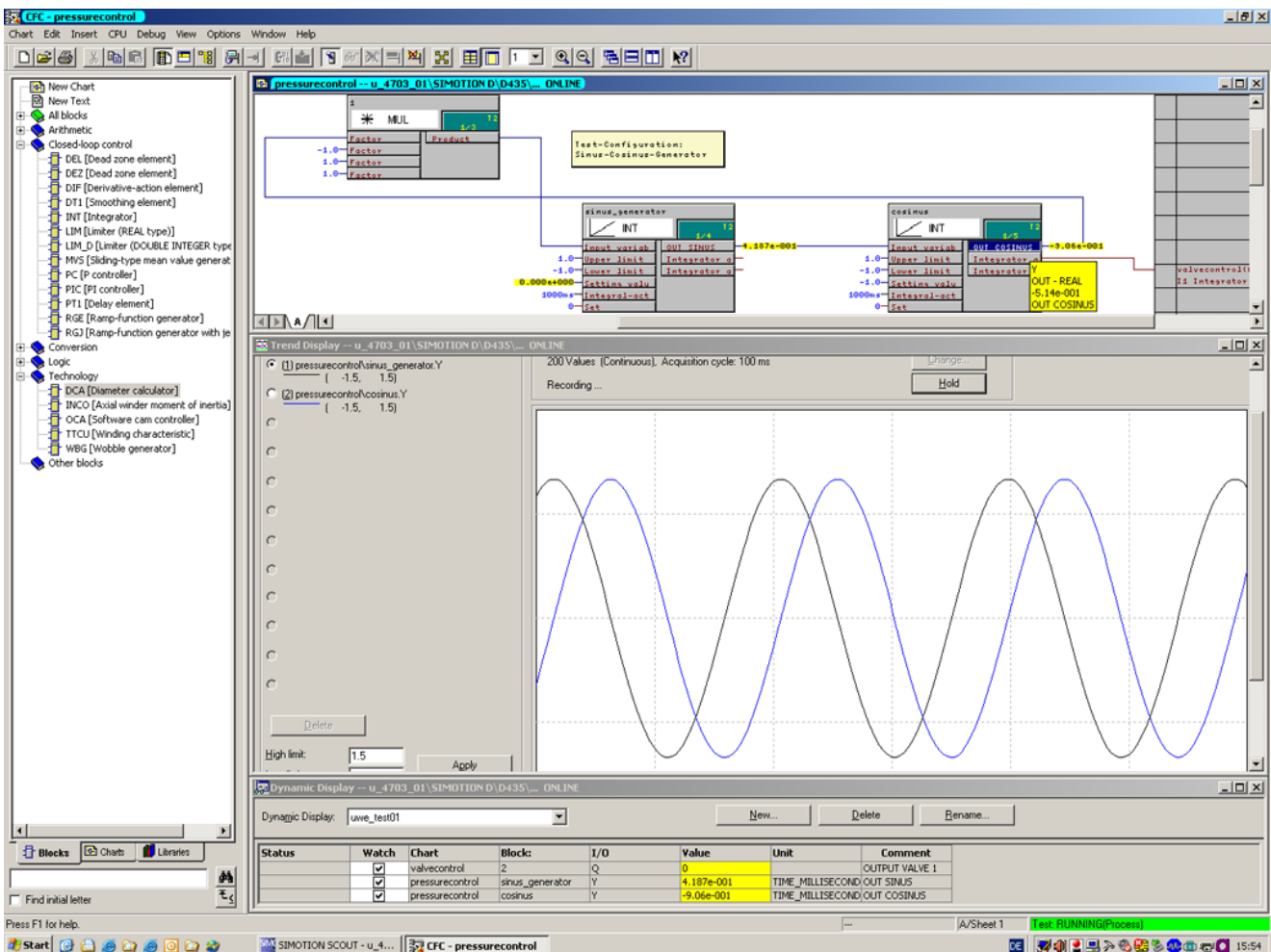
- With SINAMICS, DCC can be activated simultaneously on several drive objects (DOs) on a drive unit. With SIMOTION, several DCC charts can be created in a program container.

- Block library with management, arithmetic, control, logic and complex blocks.

- Graphic component connection editor with various editing, macro, help, search, comparison and print functions.

- Simple configuration of axial winder functions, PI controller, ramp-function generator or sweep generator.

- Execution environment for SIMOTION with sampling times that can be selected and mixed, and consistent data transfer between sampling times.

- Execution environment for SINAMICS with the option of embedding technology in SINAMICS using BICO technology so that applications can be set via configured parameters. Up to ten different sampling times can be configured.

- Diagnostics environment with signal display, diagnostic and trace functions.

- Scalability with different performance features and quantity structures in DCC SIMOTION and DCC SINAMICS.

Note

The DCC editor is a programming system based on CFC (Continuous Function Chart). The following sections describe the principles of DCC operation that have not already been covered in the manual titled "CFC Manual for S7".

Note

Further references for the DCC editor:

SINAMICS/SIMOTION Function Manual, *Description of the standard DCC blocks*

Differences between DCC SIMOTION and DCC SINAMICS

DCC SIMOTION and DCC SINAMICS differ to some extent in their mode of operation. The basic differences are listed below:

Table 2-1   Differences between DCC SIMOTION and DCC SINAMICS

	SIMOTION	SINAMICS
System integration	Via interconnection to variables of the basic system	Via adjustable parameter or interconnection via BICO parameter to the basic system
Execution system	Any number of execution groups that can be assigned five time slices (depending on the system cycle clocks)	Maximum of ten execution groups that can be assigned ten different sampling times
Consistency in the data transfer	Consistent data transfer also across the time slices	Consistency in the data transfer must be established by the user, when required, by means of standard blocks (SAH_X)
Scope of standard blocks	See SINAMICS/SIMOTION Function Manual, *Description of the standard DCC blocks*	See SINAMICS/SIMOTION Function Manual, *Description of the standard DCC blocks*

Availability of DCC features

Table 2-2   Availability of DCC features

Feature	Available as of version
Monitoring and tracing of published parameters	SINAMICS 2.5 / SIMOTION 4.1
Online changes in test mode and tracing of unpublished block connections	SINAMICS 2.6 / SIMOTION 4.1
Read back of BICO interconnections from the target device	SINAMICS 2.5 / DCC 2.0.3 / STARTER/SCOUT 4.1.3
Creation of installable libraries	SINAMICS 2.5 / DCC 2.0.3 / STARTER/SCOUT 4.1.3
DCC libraries	SINAMICS 4.4 / DCC 2.1 / STARTER/SCOUT 4.2
Installation of block libraries in the case of an opened SCOUT	SCOUT V4.1.2
Installation of block libraries in the case of an opened STARTER	STARTER V4.2
Online insertion and deletion of block instances	SINAMICS 2.6 / SIMOTION 4.1.2
Online changes, insertion and deletion of interconnections	SINAMICS 2.6 / SIMOTION 4.1.2
Display of reference data for DCC charts	SCOUT 4.1 / STARTER 4.1.3
Search and replace for DCC sheet bar interconnections	SCOUT/STARTER 4.1.2
Compiling without DCC license	SCOUT/STARTER 4.1.3
Creating C-block libraries automated from DCC libraries	DCC SIMOTION 2.0.3
Fixed execution group "BEFORE actual position value"	SINAMICS 4.3
Chart-by-chart XML export/import	SCOUT/STARTER 4.2 / DCC 2.1
User-defined structures, DCC SIMOTION	SCOUT 4.2 / DCC 2.1
Read back DCC chart sources from the target device	SINAMICS 4.4 (only in conjunction with TPdcblibV3.0_SINAMICS_4_4 or higher) / SCOUT / STARTER 4.2 / DCC 2.1
Automatic library exchange when upgrading the device version	SCOUT/STARTER 4.2
Exchanging target device families for DCC libraries	SCOUT 4.3 / DCC 2.2

DCC editor functionality

<div align="right">3</div>

3.1 Overview

The product offers a modular, scalable technology option, which has chiefly been developed for drive-related, continuous open-loop and closed-loop control engineering tasks.

The DCC technology option for SIMOTION controllers and SINAMICS drives can be configured graphically using the Drive Control Chart editor (referred to below as DCC editor), which is based on CFC. The following figure illustrates the data flow of the configuration data when configuring with the DCC technology option:
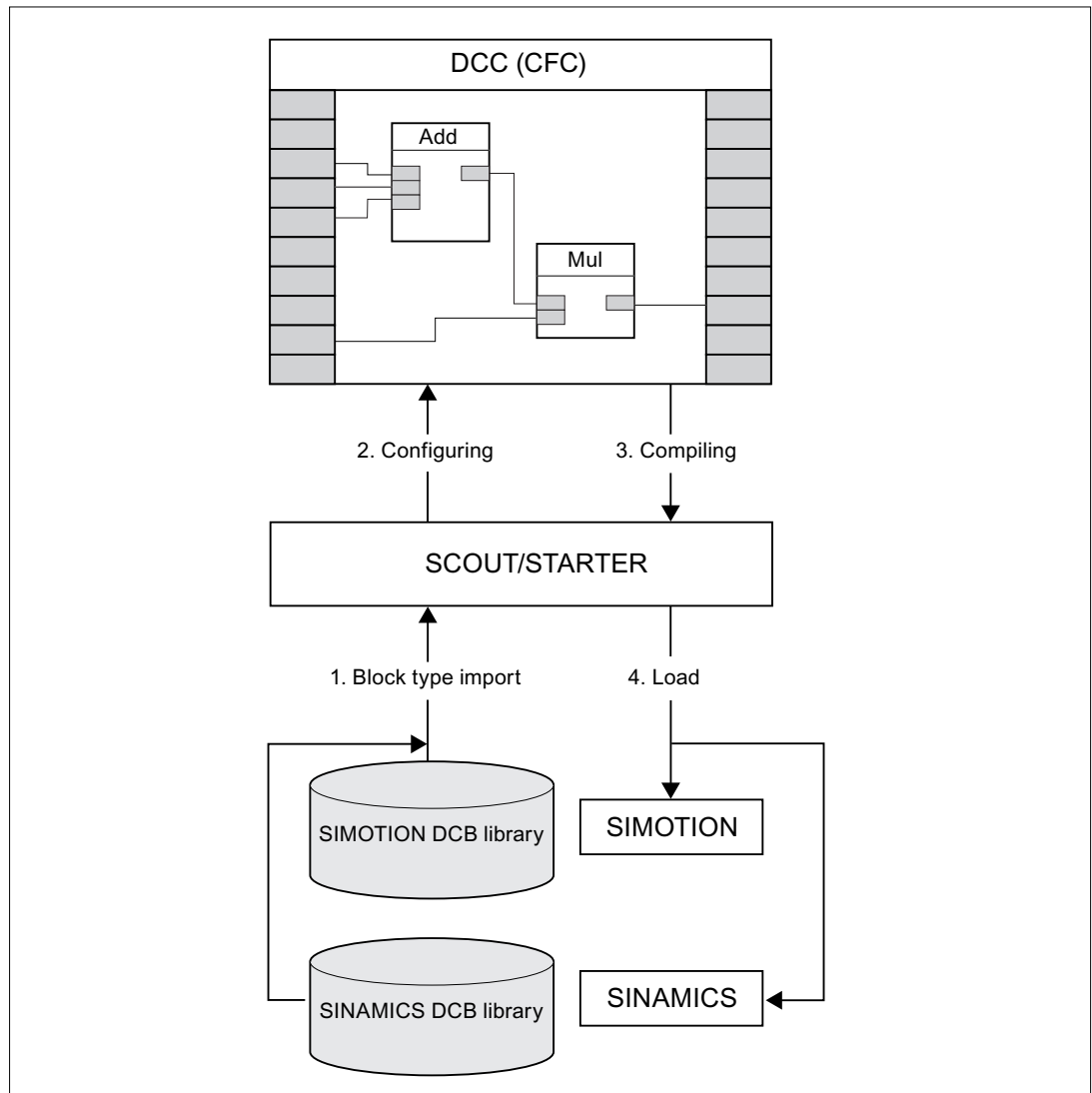
Figure 3-1   Flow of configuration data

1. When a new chart is created, the block types are taken from the device-specific block library and inserted in the DCC editor's block manager.

2. The DCC editor is used to create charts, in which you can insert, parameterize and interconnect blocks.

3. When you compile the charts, an intermediate code is created.

4. This is downloaded to the device or drive unit using STARTER/SCOUT.

3.2 Requirement

This description refers to the following device SW versions:

- DCC 2.3

- SIMOTION P, C, and D, as of Version 4.4

- SINAMICS Integrated of the SIMOTION D as of Version 4.7

- SINAMICS S120, S150, SM150, G130, G150, GM150, GL150 as of Version 4.7

- SINAMICS DCM as of Version 1.4
  Except for the utilization display, the SINAMICS DC MASTER as of Version 1.4 supports the same function scope as SINAMICS S120 as of Version 4.7.
  Detailed information about *SINAMICS DC MASTER* Version 1.4 can be found in the Operating Instructions *SINAMICS DC MASTER 6RA80 DC Converters*, 01/2014 Edition

- SIMOTION SCOUT/STARTER as of Version 4.4

- At least CFC 7.0.1.3 must be installed as of DCC 2.0.3.

- At least CFC 7.1. must be installed as of DCC 2.1.

- DCC 2.2 can be used with the following licenses:

  - STARTER: CFC 7.0.1.z, z => 3, CFC 7.1.x, x >=2 and CFC 8.0

  - SCOUT: CFC 7.1.x, x >=2 and CFC 8.0

- DCC 2.3 can be used with the following licenses:

  - STARTER: CFC 7.0.1.z, z => 3, CFC 7.1.x, x >=2 and CFC 8.0 and CFC 8.1

  - SCOUT: CFC 8.0.x, x >=2 and CFC 8.1

Note

If SCOUT is reinstalled and the CFC is not installed along with it, you can continue to use the version of CFC that is already installed.

Note

If errors occur when charts are being compiled in SIMOTION SCOUT/STARTER, it is possible to generate a comprehensive error report by compiling them again in the DCC editor. To do this, activate **Display all messages with 'Save and compile changes'** on the **Compiler** tab under the **Settings** menu command in SIMOTION SCOUT. A corresponding error report is automatically created in STARTER.

Note

The DCC editor is installed automatically with the SIMOTION SCOUT/STARTER engineering system.

DCC setup 2.0.2 - 2.0.5 can be installed with all SIMOTION SCOUT/STARTER V4.1 service packs and hotfixes.

DCC setup 2.0.2 - 2.0.5 can be installed in SIMOTION SCOUT/STARTER 4.2.

DCC setup 2.1 can be installed as of SIMOTION SCOUT/STARTER 4.2.

DCC setup 2.2 can be installed as of SIMOTION SCOUT/STARTER 4.3.

DCC setup 2.3 can be installed as of SIMOTION SCOUT/STARTER 4.4.

An appropriate program license is required for the DCC editor. This is on the USB stick supplied with the DCC SIMOTION or DCC SINAMICS product.

Note

It is only possible to print DCCs in the DCC editor, but not in SIMOTION SCOUT/STARTER.

Note

Only context-sensitive help is available for the DCC blocks, i.e. the descriptions cannot be accessed from the main help page.

Note

If the DCC charts are opened with a different CFC version than the one with which they were created, this may lead to inconsistencies within the project. The DCC chart must be recompiled and reloaded. DCC charts that have been edited with a newer CFC version can no longer be opened with an earlier CFC version. Backward conversion is not possible.

If you open a project whose charts have been created with an older CFC version, you can select whether the data format of the charts should be updated. If you reject this, you can in fact view the CFC charts, but cannot edit them any further. It is recommended to adapt the data format of the CFC charts to the currently installed CFC version.

3.3 New device versions

SIMOTION

The DCC functionality and performance of the new SIMOTION target devices is compatible with the previous versions, i.e. DCC V2.3 within the framework of SIMOTION SCOUT 4.4 supports the following target device versions:

- SIMOTION V4.1 devices
- SIMOTION V4.2 devices
- SIMOTION V4.3 devices
- SIMOTION V4.4 devices

The DCC functionality and performance of the new SOC2-based SIMOTION target devices (e.g. D410-2 DP) is compatible with the corresponding IDC-based previous devices and versions (e.g. D410 DP).

Note

A compatibility list can be found on the SIMOTION SCOUT DVD and on the Internet at the following link:
http://support.automation.siemens.com/WW/llisapi.dll?
func=cslib.csinfo&lang=de&objid=18857317&caller=view

SINAMICS

DCC SINAMICS supports all previously released device versions with DCC in STARTER/ SCOUT V4.4.

In addition, DCC is offered for:

- SINAMICS 4.7
- DCM 1.4
- DO type HLA.

The DCC functionality and performance of the new SOC2-based SINAMICS target devices (e.g. CU320-2 DP) is compatible with the corresponding previous IDC devices and versions (e.g. CU320 DP).

The following device types / device versions are supported with SINAMICS 4.7:

- S120 CU320-2
- S120 Integrated
- CX32-2
- S150 CU320-2
- GL150 CU320-2
- GM150 CU320-2
- SL150 CU320-2

- SM120 CU320-2
- SM150 Integrated

Note

With the installation of an SSP, the latest standard *dcblib* library available for the device is installed and can be used on the device.

3.4 Establish the project requirements

3.4.1 Create a project

You must create a new project in the SIMOTION SCOUT/STARTER engineering system before using the DCC editor.

Procedure

Proceed as follows to create a project:

1. Start the SIMOTION SCOUT/STARTER engineering system.

2. Execute the **Project > New** function from the menu bar.

3. In the **New Project** window, enter the name of the project in the **Name** field.
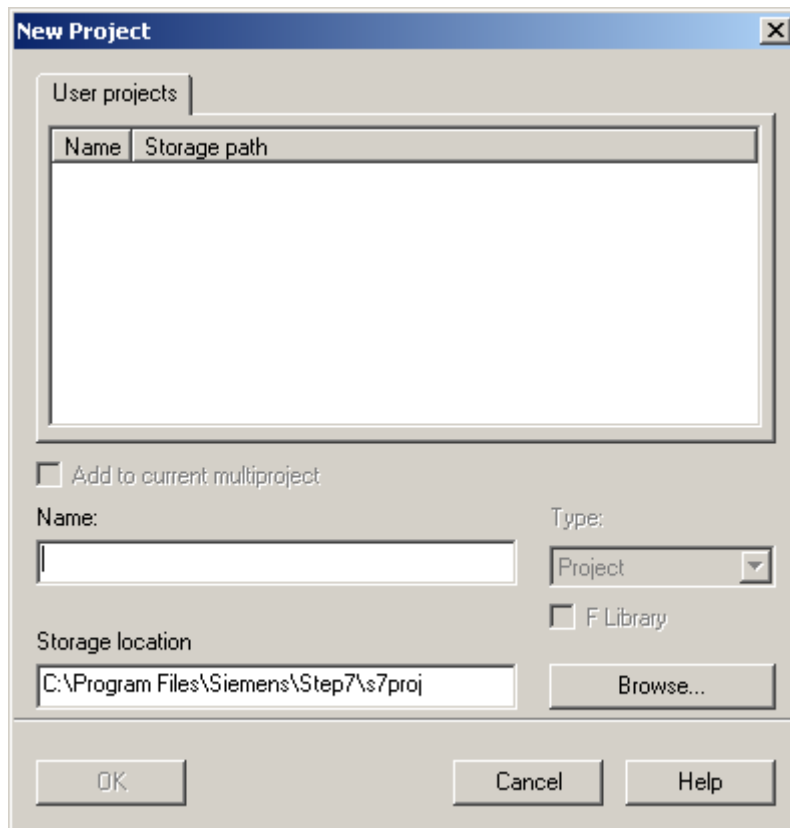
Figure 3-2   Window - New Project

4. Click **OK** to close the window.

The new project is created and then automatically opened.

Note

Convention for assigning names for projects

The project name may contain a maximum of 24 characters. The folder name is generated using the first eight characters of the project name. It is therefore important to ensure that the first eight characters of the project name are unambiguous.

3.4.2 Inserting a device into a project

SIMOTION

Proceed as follows to insert a device into a project:

1. Open an existing SIMOTION SCOUT project if a project is not already open.

2. Execute the **Create new device** command.

```
☐ DCC_1
    ▶ Create new device
    ▶ Insert single drive unit
    ☐ LIBRARIES
        ▶ Insert library
        ▶ Insert DCC library
    ☐ MONITOR
```
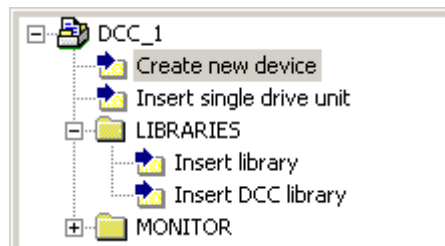
Figure 3-3   SIMOTION: Create new device

3. In the **Create new device** window, select the required device and close the window with **OK**.

Note

For further information on the **Open HW Config** switch, please refer to the documentation for SIMOTION SCOUT.

All requirements for creating a DCC in the project have now been met.

SINAMICS

1. Open an existing project or create a new project in which you want to insert a SINAMICS drive unit (e.g. SINAMICS S120 CU 320). Note that the SINAMICS S110 (CU 305 module) does not support DCC.
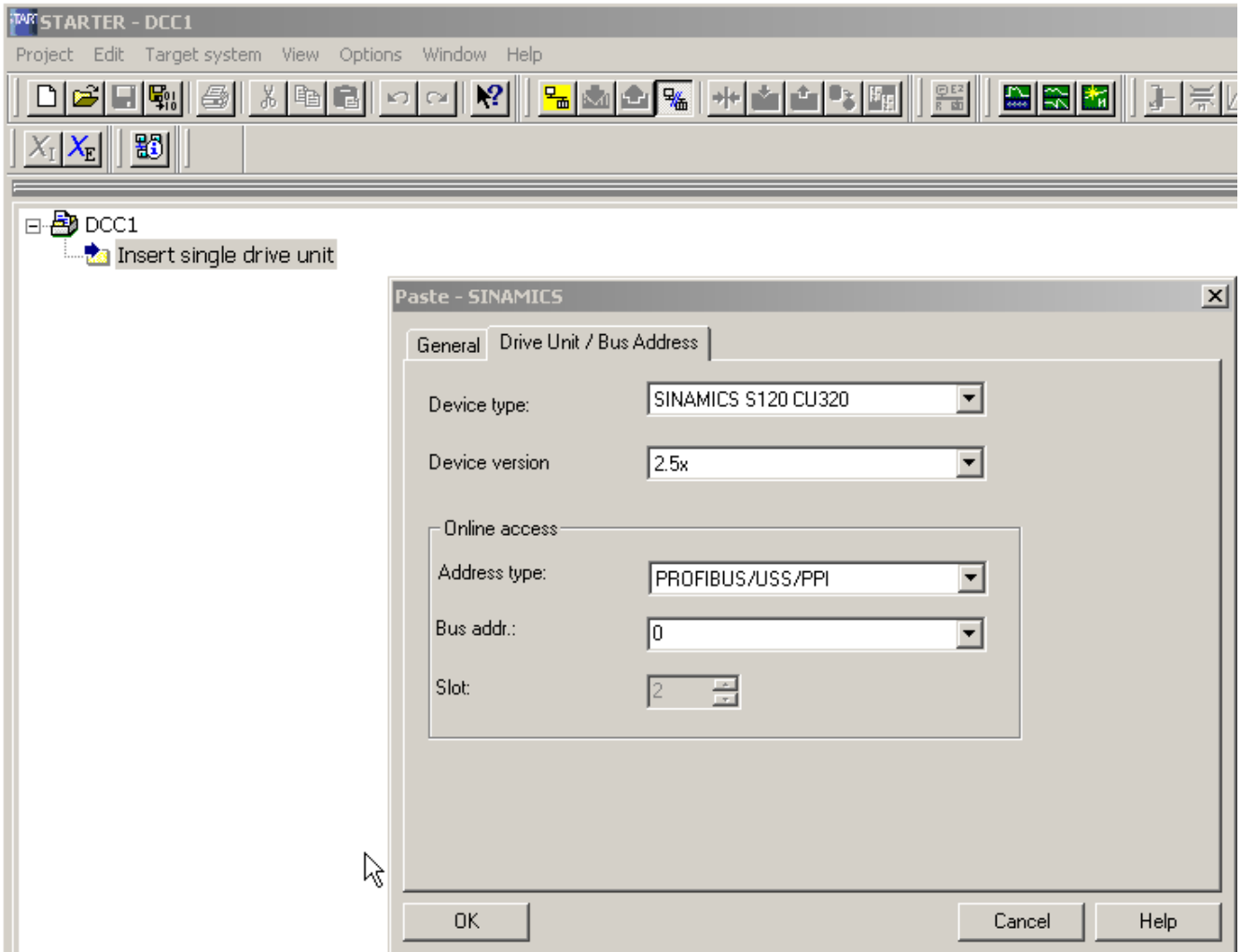
Figure 3-4   Inserting a device into a project

3.4.3   Inserting the DCC chart in a project

You can now insert a DCC chart into the existing project.

Procedure

1. Select a device from the project.

2. **SIMOTION:**
   Execute the **Insert DCC chart** function from the **PROGRAM** subitem of the device.
   SINAMICS, STARTER:
   Execute the **Insert DCC chart** function on the desired drive object. There may only be one DCC chart on a drive object.
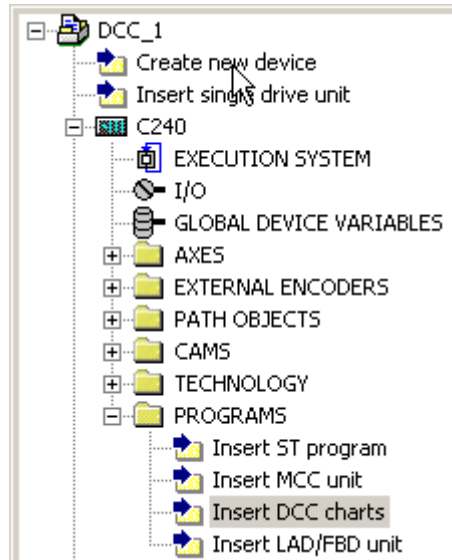
Figure 3-5   SIMOTION SCOUT: Inserting a DCC chart
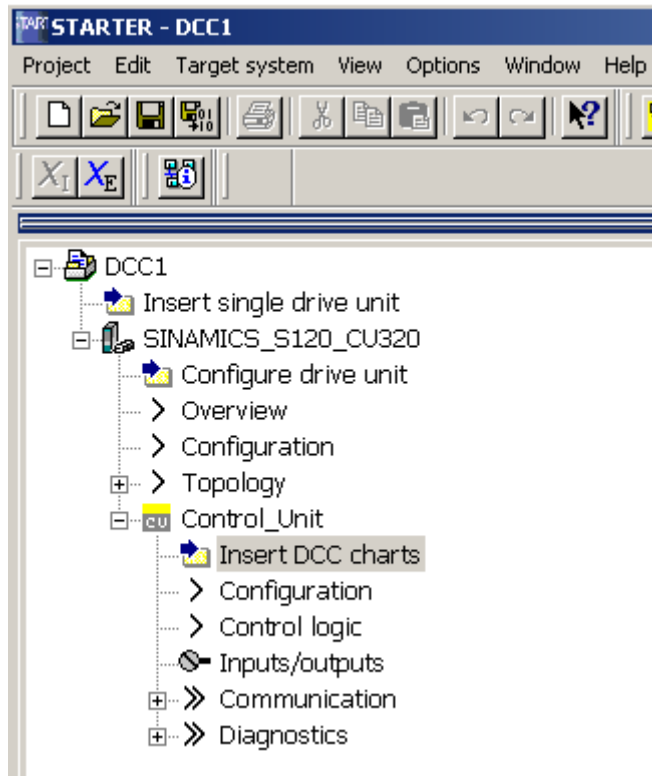
Figure 3-6   Inserting a DCC chart in the CU drive object of a SINAMICS CU3x0.x drive unit with STARTER
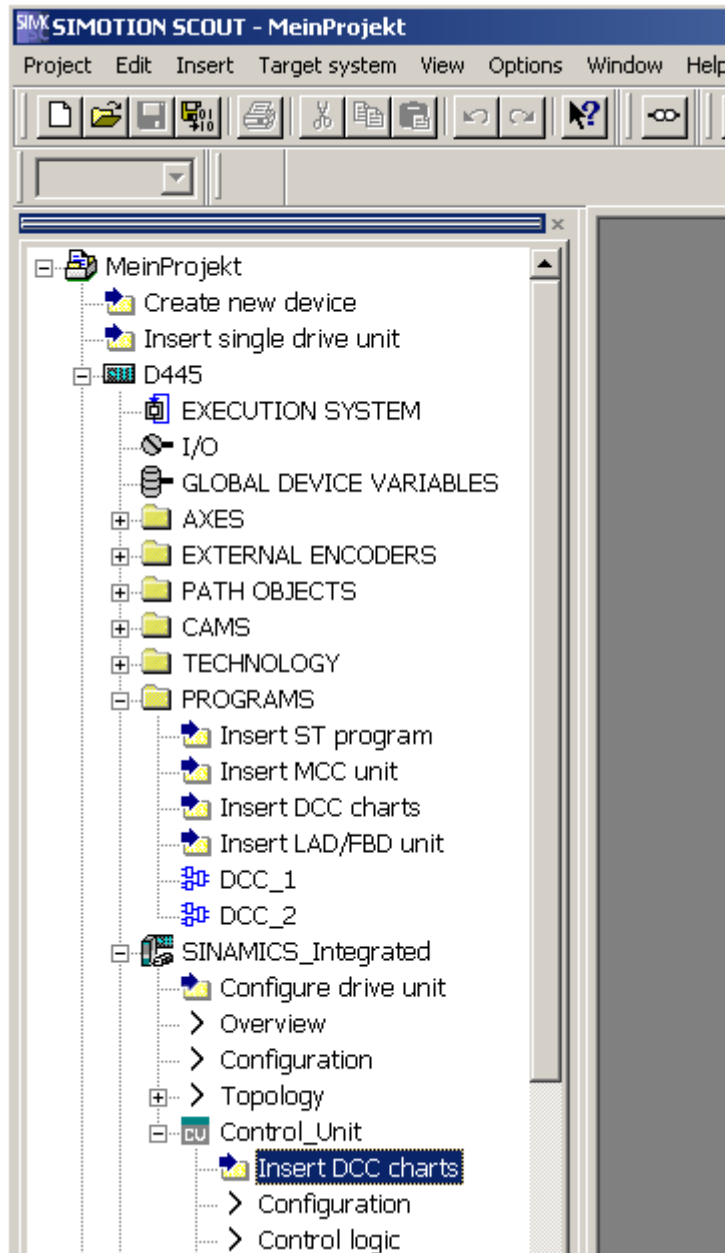
Figure 3-7   SIMOTION D4xx: Inserting a DCC SINAMICS chart on the drive object of the CU with SCOUT

3. The **Insert DCC Chart** window appears.
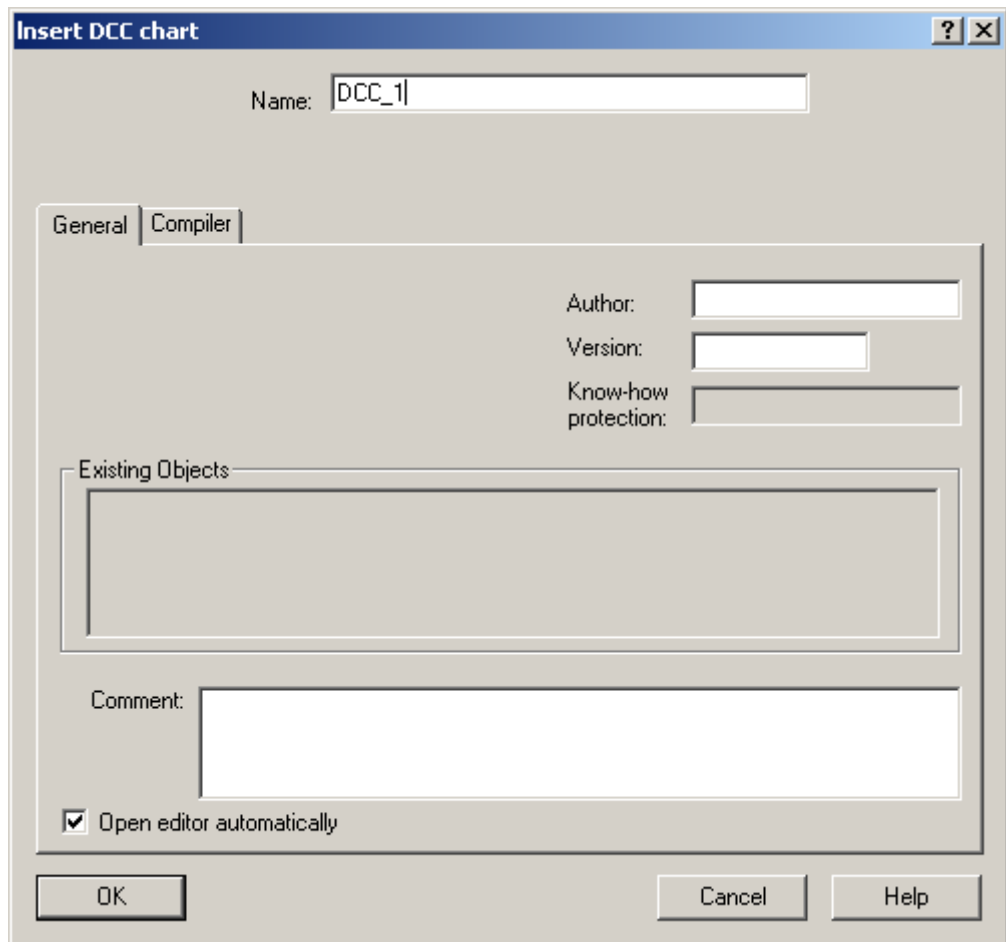
Figure 3-8   Window - Insert DCC Chart

4. Assign a name to the DCC chart.

The DCC chart has now been created.

If you have selected the **Open editor automatically** option in the **Insert DCC Chart** window then the DCC editor opens automatically. When opening the DCC chart for the first time, the **Import DCB Library** window is automatically displayed.

Note

Convention for assigning names to charts

The chart name may contain a maximum of 22 characters.

Explanation of the various types of chart

There are three different types of chart:

- Basic chart
- Chart partition
- Subchart

Charts that are visible within SIMOTION SCOUT/STARTER or the SIMATIC Manager are designated as basic charts. Every basic chart has up to 26 chart partitions and each of these partitions comprises six sheets. Embedded charts - the subcharts - can be used within each sheet. Each of these subcharts may also have its own chart partitions and subcharts. A maximum number of eight nesting levels with subcharts is possible.

Subcharts are not visible as charts in SIMOTION SCOUT/STARTER or in the SIMATIC Manager.

The following graphic clarifies the connection between the three types of chart.
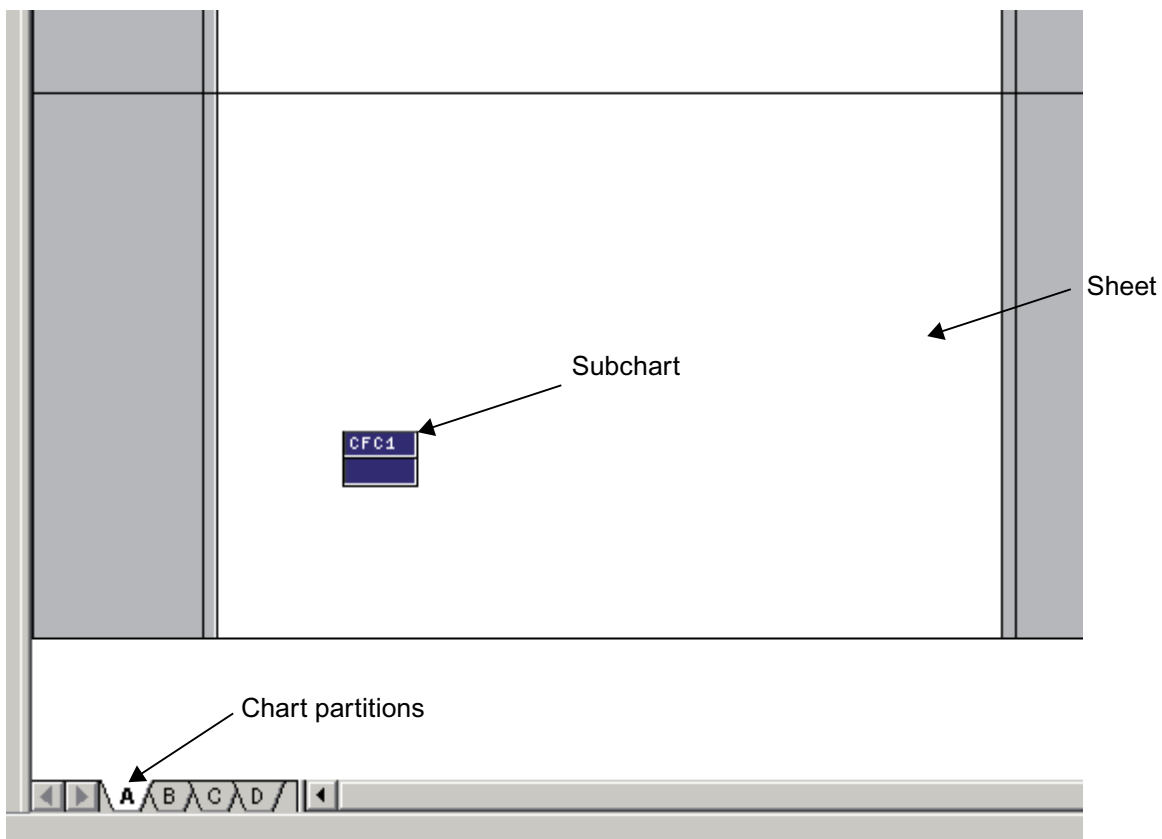
Figure 3-9   Connection between the types of chart in the DCC editor

Select the DCC chart in the project navigator and open the **DCC Chart Properties** dialog via the **Properties...** context menu to access certain properties of the DCC chart. On the **General** tab under **Time stamp** under **Last changed on (STEP7):** you can see the date on which the DCC chart was changed the last time by STEP 7 (CFC editor) or changes accepted in **DCC**

chart properties. Under **Last changed on:** you can see the date on which the DCC chart was last compiled or changes accepted in **DCC chart properties**. Under **Project storage location:** you can see the path under which the project was stored.
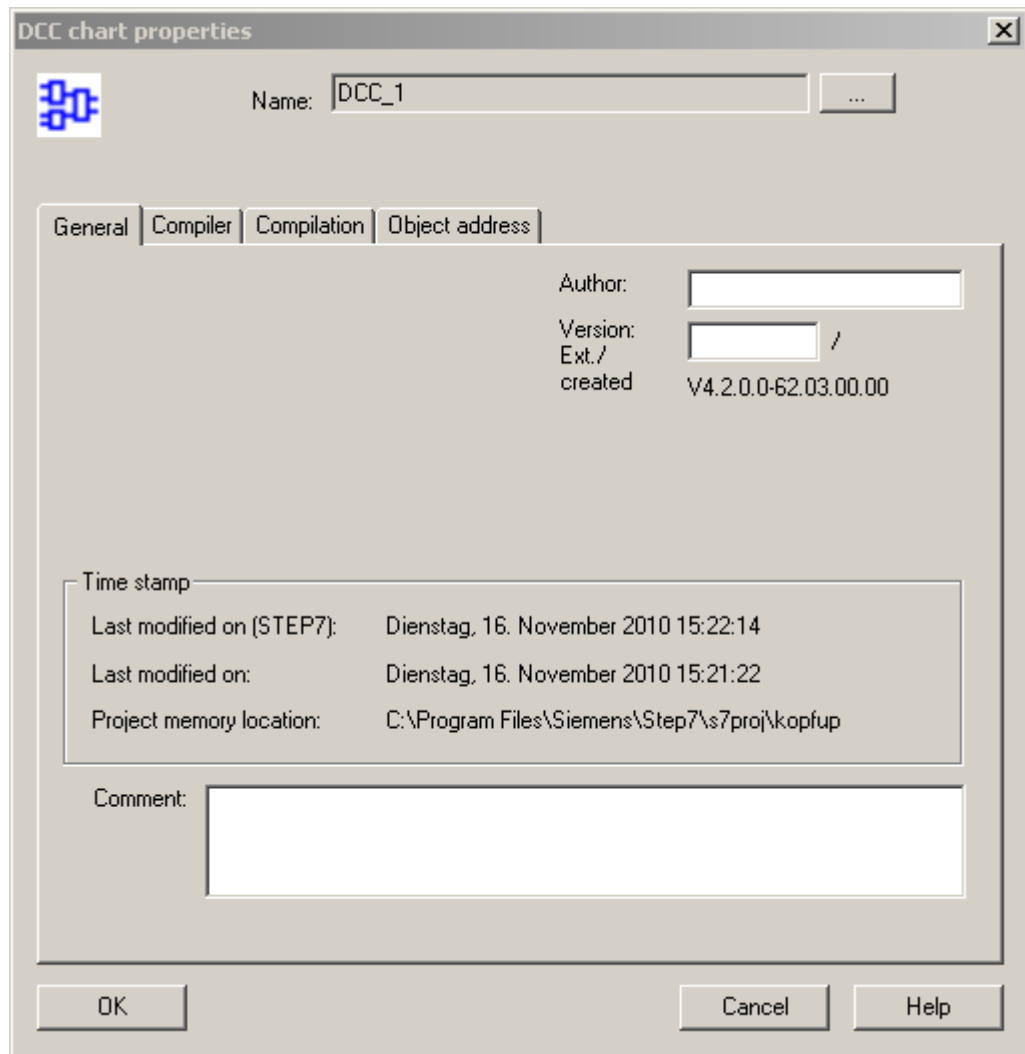
Figure 3-10   DCC chart properties

3.4.4   Inserting a new subchart

A chart (subchart) can be inserted in another chart (chart-in-chart-technique). Hierarchical structures can be formed here. Each chart that is inserted can be opened and modified. A chart can be encapsulated for further use, i.e. chart I/Os added. Which block connections are provided at the chart connections can also be specified individually.

Requirement

You have created a DCC chart in the SIMOTION SCOUT/STARTER engineering system which is opened in the DCC editor.

Procedure

1. Use the **View > Overview** menu command or the ⊞ button on the toolbar to switch from the page view to the chart overview. The six pages of the selected DCC chart are shown.

2. Use the **Insert new chart** context-menu command to insert a new subchart, then open the subchart by selecting **Open** from its context menu.

Note

DCC charts should always be created in STARTER/SCOUT.

If a DCC chart that is assigned to a SINAMICS drive object is open and additional DCC charts are created directly in the CFC editor, this can result in compilation errors.

3.4.5 Inserting new chart partitions

Requirement

You have already created a DCC in the SIMOTION SCOUT/STARTER engineering system which is opened in the DCC editor.

Procedure

1. Insert a new chart partition in the desired position using the menu items **Insert > Chart partition > In front of current chart partition** or **At the end**.

2. Alternatively, you can right-click an already existing chart partition on the tab and select **Insert chart partition in front of current chart partition** or **Insert chart partition at the end**.

3.4.6 View and representation

Going into the page view or overview representation

To change to the page view from the overview representation, right-click an empty space in the chart and select **Display this page** in the context menu that appears. The names of the block connections are displayed in this enlarged view.

To change to the overview representation from the page view, right-click an empty space in the chart and select **Overview** in the context menu that appears.

You can also switch to the page view and back to the overview again by double-clicking an empty area on a page.

It is also possible to switch between the page view and overview representation using the **View** menu.

You can switch to a page view with the block catalog displayed on the left using the **View -> Catalog** menu.

3.5 Handling blocks

3.5.1 Introduction

In this section, you will learn which block types are available and how you can insert blocks in a DCC and delete them. You will also learn how to edit block connections.

Note

The online help provides detailed information (incl. timing diagram and plant view) for the individual blocks. To start the help, select the required block in the chart or in the block catalog and press the F1 key.

3.5.2 Inserting blocks in the DCC editor

Block types in the DCC editor

The block type inventory featured in the block catalog depends on both the device type and the version of the library. You will find the directories for the block families, as well as the directories **All blocks** (containing all blocks) and **Other blocks** (blocks that are not assigned to a family), in the block catalog. The names of the block families in the DCC editor are always in English.

Inserting a block
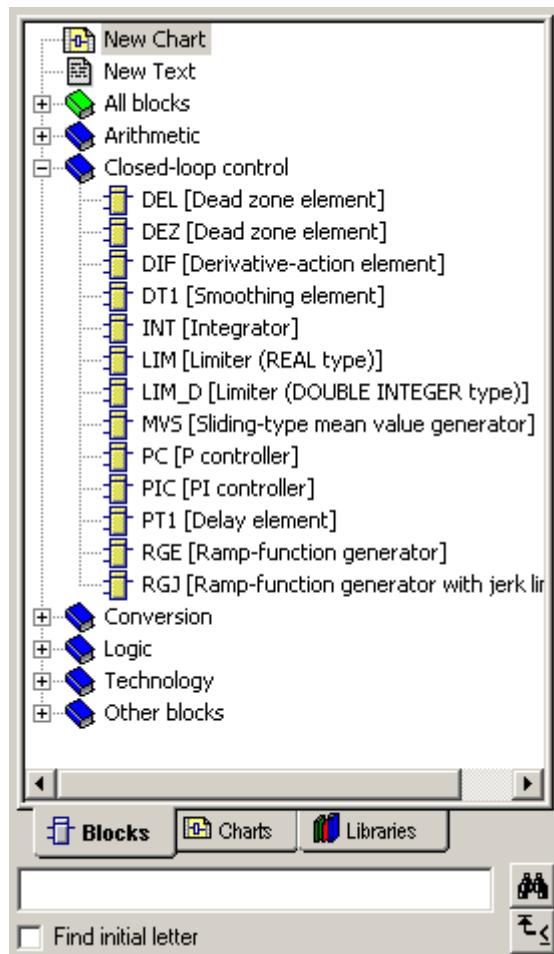
Figure 3-11   Inserting blocks

● Open a block family in the **Closed-loop control** family with the closed-loop control blocks.

● Select the required block and insert it in the chart using drag-and-drop. Only the outline of the block in dashed lines is displayed during the copying procedure. Release the mouse button at the required point.

● To search for a block, enter its name in the input field of the block catalog and click the **binoculars** button. The search process begins.

Note

If blocks are superimposed on the chart with other elements, such as other blocks or the sheet bar, the superimposed block will be displayed in gray and its connections will not be visible. You must reposition the blocks to ensure that all block information can be viewed.

3.5.3   Inserting text

You can add comments to your DCC that you enter in text fields. You can place these at any free position in the chart.

Procedure

Select the **New text** command via the directories of the block types and insert it in the chart using drag-and-drop. Release the mouse button at the required point.
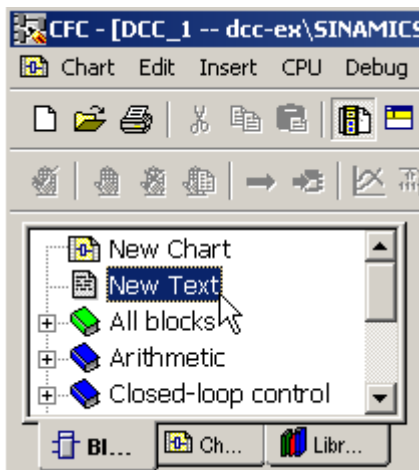
Figure 3-12   Inserting a text field

Alternatively, you can right-click at the desired position in the chart and select the **Insert new text** command in the context menu. Note that this option is only available offline.
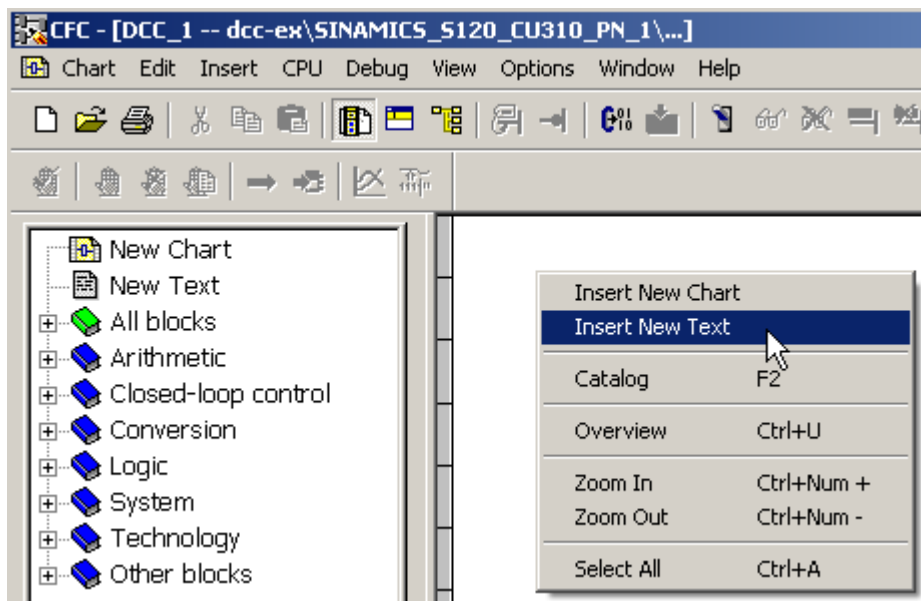
Figure 3-13   Inserting new text

You can adapt the size of the text field to your requirements by selecting the black points at the corners and the sides and dragging them to the desired size.

You can change your comments by clicking the text field and then entering or editing your text.

3.5.4 Specifying execution properties

You can display or change the execution properties of all the used blocks of the program. You can display the properties in the toolbar via **Edit > Execution sequence** or via the 🔳 button.
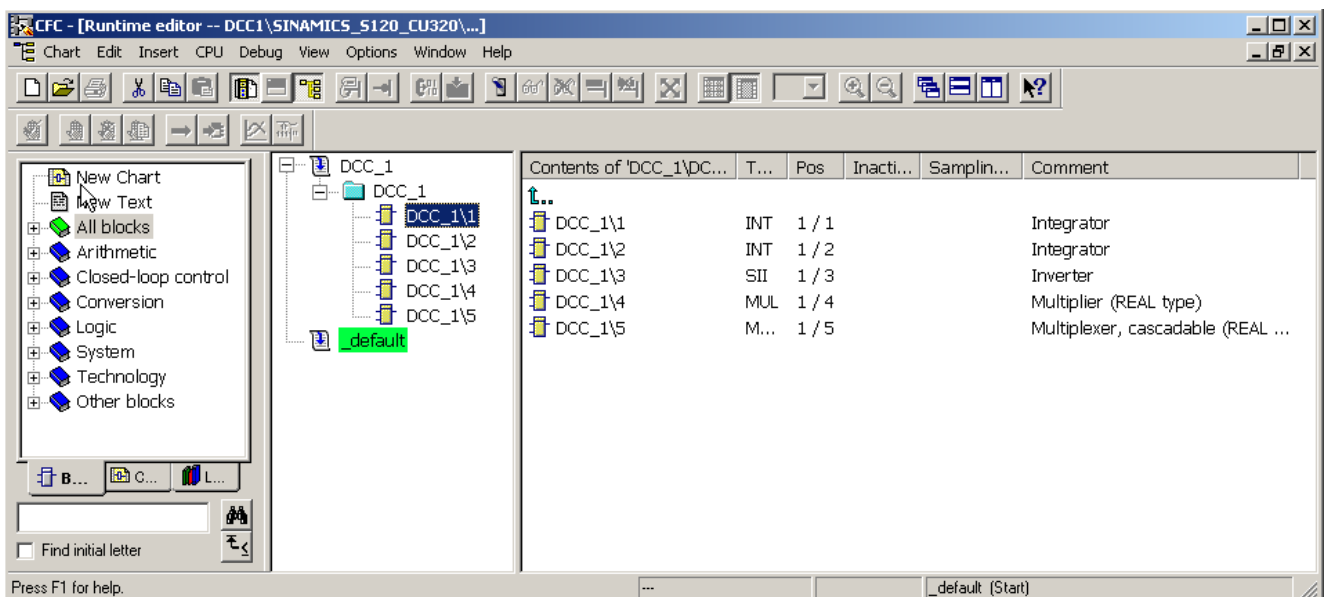
Figure 3-14   Execution editor of the dccReg1 chart with the execution groups Tsg_dccReg1 and Tsg2

In this window, you can also change the insert point in the execution sequence by dragging the block to the desired position. The assignment of a block to an execution group can be changed in the same way.

New inserted blocks are always placed in the execution system behind the block that is defined as predecessor. By default, this is always the block that has been inserted last. If a block is to be inserted in the execution sequence behind an already existing block, select the block with the desired offset in the overview, right-click and select the **Predecessor for insert point** function in the menu. The block now defined as predecessor block is displayed in light green in the DCC editor.
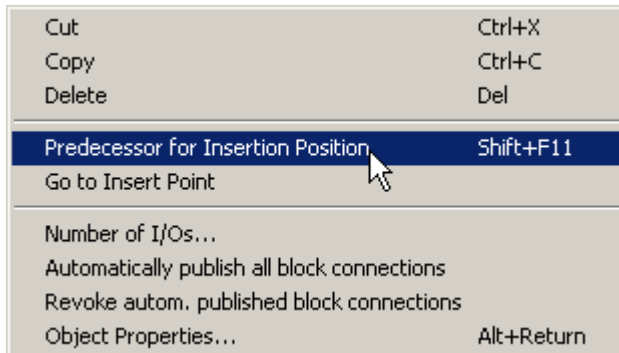
Figure 3-15   Block context menu: Defining predecessor for insert point

The active execution position is displayed in the status bar. This is at the bottom right in the execution editor.

3.5.5   Editing block connections

3.5.5.1   General

There are two types of block connection (inputs and outputs), each of which has a distinct function and is edited in a particular way.

The following sections contain further information about handling the block connections in the DCC.

3.5.5.2   Block connection properties

You can double-click each of the individual connections to parameterize it. Alternatively, the **Properties - Connection** window can also be displayed as follows:

1. Select the desired connection.

2. Select **Object properties** in the context menu.

3. The **Properties - Connection** window appears.

However, it is easier to parameterize the inputs as follows:

- Double-click a block header. The **Properties - Block** window appears. You can also open the **Properties - Block** window via the **Object properties** context menu command of the block or via the **Edit > Object properties** menu command.

- Click the **Connections** tab. The parameters in fields with a gray background cannot be changed.

- Enter the required values in the table and click **OK** to close the dialog box.

Input values

At the block inputs, a value can be entered in the **Value** field of the properties dialog box. If the input is not interconnected, it always has the specified value. With interconnected blocks, the output value of the upstream block always applies in the initialization phase and in the first cycle.

Output values

At the block outputs, a value can be entered in the **Value** field of the properties dialog box. In the first cycle however, the specified value is overwritten by the calculated value.

Note

Special feature with hidden block connections

In the DCC editor, you can hide block connections to improve the clarity of the configured charts. However, the hidden block connections remain active in the DCC, so their values are still evaluated. With generic blocks, you can also reduce the number of block connections in the DCC editor. The hidden connections are assigned default values. However, the hidden block connections remain active in the target system, so their values are still evaluated.

It should also be noted that the inputs of these blocks must be interconnected consecutively, starting from the first connection.

Number of block inputs

The number of inputs could be increased for the AND, ADD, MAS, MIS, MUL, NAND, NOR, OR and XOR blocks from the standard library. However, the DCC editor can only evaluate four input signals per block, and therefore this is not permitted. If the demand is greater, the block must be called several times. The note above also applies to data-type-specific variants of the blocks listed above.

Pseudo comments

Comments at block connections that start with @ are pseudo comments; they influence the function of the block connection and serve as interfaces to the basic system.

For further information about the pseudo comments, see:

● Creating customer-specific parameters ("declare") (Page 156)

Block connection units

The block connection units that can be set in the Properties dialog box serve only as comments in the DCC editor - the values are not used for automatic conversions.

Configuring the block display

You can change the display of the blocks. You can change the block width via **Options > Settings > Block / sheet bar width.....**.

You can change the names of the block inputs/outputs via **Options > Settings > Display** in the submenu **Connections**.

The block type can be displayed in the form of both text and graphics. This can be configured via **Options > Settings > Display** in the submenu **Block headers**.

If you want to display more than the first eight characters of the comment, then select **Options > Settings > Block / sheet bar width**. Then in the **Blocks / Sheet Bars** window, set the block width to **Wide**. The first twelve characters of the comment are now displayed at the connection.

See also

@ variables (SIMOTION) (Page 207)

3.5.6 Interconnecting blocks

Blocks can be interconnected with one another. The outputs of a block then form the inputs for further blocks.

Requirement

The inputs and outputs of the blocks must possess compatible data types so that they can be interconnected. An overview of which data types can be interconnected is shown below:

Table 3-1   Conversions

Input	Output	Description
WORD	INT	Interconnection of a word variable to an integer variable
INT	WORD	Interconnection of an integer variable to a word variable
DWORD	DINT	Interconnection of a double word variable to a double integer variable
DINT	DWORD	Interconnection of a double integer variable to a double word variable
BYTE	SINT	Interconnection of a byte variable to a short integer variable
SINT	BYTE	Interconnection of a short integer variable to a byte variable
USINT	BYTE	Interconnection of an unsigned short integer variable to a byte variable
BYTE	USINT	Interconnection of a byte variable to an unsigned short integer variable
USINT	SINT	Interconnection of an unsigned short integer variable to a short integer variable
SINT	USINT	Interconnection of a short integer variable to an unsigned short integer variable
UINT	WORD	Interconnection of an unsigned integer variable to a word variable
WORD	UINT	Interconnection of a word variable to an unsigned integer variable
UINT	INT	Interconnection of an unsigned integer variable to an integer variable

Input	Output	Description
INT	UINT	Interconnection of an integer variable to an unsigned integer variable
UDINT	DWORD	Interconnection of an unsigned double integer variable to a double word variable
DWORD	UDINT	Interconnection of a double word variable to an unsigned double integer variable
UDINT	DINT	Interconnection of an unsigned double integer variable to a double integer variable
DINT	UDINT	Interconnection of a double integer variable to an unsigned double integer variable
SDTIME	REAL	Interconnection of an SDTime variable to a real variable

Procedure

Connect the output of the first block (source) with the input of the second block (sink). You can create this connection either by using drag-and-drop or by latching them together (clicking the relevant input connection and output connection once).

The connection line is automatically drawn from the output of the first block to the input of the second block.

3.5.7 Data type abbreviation in the DCC for connection and transformer blocks

Table 3-2   Table of data types

Abbreviation	Keyword	Name	Bits
BO/B	BOOL	Logical number	8
BY	BYTE	Sequence of 8 bits	8
DI/D	DINT	Double integer	32
DW	DWORD	Sequence of 32 bits	32
I	INT	Integer	16
PC	LREAL	Double floating-point number Accuracy according to IEEE754	64
R	REAL	Floating-point number	32
SI	SINT	Signed short integer	8
TS	SDTIME	The SDTIME data type is derived from the REAL data type; 1.0 corresponds to 1.0 ms Negative values are not defined.	32
UD	UDINT	Unsigned double integer	32
UI	UINT	Unsigned integer	16

Abbreviation	Keyword	Name	Bits
US	USINT	Unsigned short integer	8
W	WORD	Sequence of 16 bits	16

3.5.8 Interconnection to chart connections

A chart can be encapsulated for further use, i.e. chart connections added. Which block connections are provided at the chart connections can also be specified individually.

Procedure

1. Model your DCC in the DCC editor.

2. Select the Chart Connections window via **View** -> **Chart connections** or with the button. The chart connections (IN, OUT, IN_OUT) are shown in this window.

Note

Please note that chart connections of type IN_OUT are not permitted in DCC!
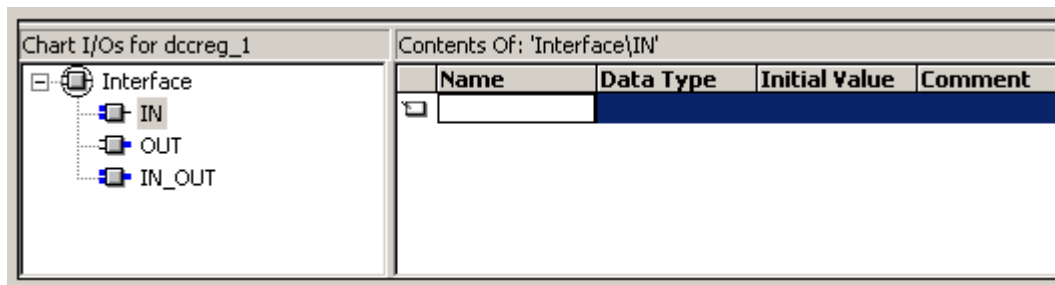
Figure 3-16   Screenshot showing chart connections area of the window

3. Define the required chart connections.

4. Interconnect the block connection to the chart connection. There are three ways of doing this:

   – Option 1:
   Drag the block connection using drag-and-drop to IN (for inputs) or to OUT (for outputs) and keep the Ctrl key pressed. A chart connection is now created automatically.

   – Option 2:
   Select the interface type (IN or OUT) from the left pane and then drag the desired block connection into the last (empty) line in the right pane using drag-and-drop.

   – Option 3:
   Right-click the block connection to be interconnected and select **Interconnection to the chart interface** in the context menu. Then select the appropriate chart connection in the **Insert/Change Interconnection to the Chart Interface** window. Click OK to close the window.

   Note

   The interconnection to the chart connection by the context menu is available only when at least one chart connection exists already. Up to CFC 7.0, the first interconnection to the chart I/O can only be established using drag-and-drop.

The block connection is interconnected to the chart connection. The assignment is displayed in the variables sheet bar of the DCC editor. The block interface is defined as part of assigning block connections to the interface.

Note

DCC SINAMICS: The use of chart I/Os is permitted for subcharts and DCC libraries.

3.5.9 Interconnection to global operands in DCC SIMOTION

Global operands are connection partners located outside of the DCCs.

Interconnections to global operands are entered in the sheet bar.

Where DCC is concerned, the interconnection of global operands serves as an interface to the basic system. You use this function to connect to ST/IO/system variables (in the case of DCC SIMOTION) or BICO parameters (in the case of DCC SINAMICS) in the drive.

Procedure

You can make an interconnection to a global operand as follows:

1. Open the DCC.

2. Right-click the block connection to be interconnected and select **Interconnection to operand** in the context menu.

3. Now select the global operand to be interconnected in the **DCC Signal Selection** window.

4. Click **OK** to close the window.

The block connection is interconnected to the selected global operand.

3.5.10 Deleting blocks

If you want to delete a block from the chart, select it and click **Edit > Delete**. You can also delete blocks using the context menu.

When blocks are deleted, the connections to the block connections are also removed. Output interconnections must first be deleted manually.

If you delete a block in online mode on which the outputs are interconnected with inputs of other blocks, the current signal values will become valid at the inputs of these blocks. These are taken over in the DCC and also saved to the card in the target system at the next RAM to ROM. In offline mode, the default values take effect again at the inputs after deletion of the upstream block.

Note

The STM block cannot be deleted or inserted online.

If you delete a block on which the inputs or outputs are interconnected to chart connections, the connections are deleted, but not the chart connections. If these are not required, you must delete them separately in the Chart connections field.

3.5.11 Rearranging parameter numbers

Call the **Rearrange Parameter Numbers** dialog box via the **Chart -> Rearrange parameter numbers** menu to move parameter numbers.
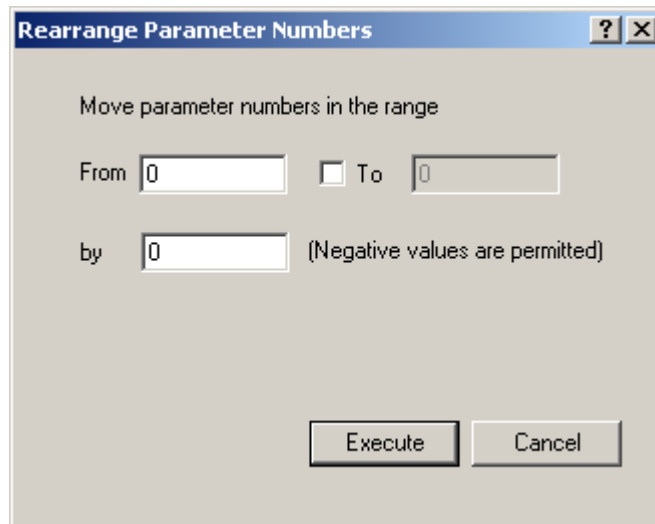
Figure 3-17   Rearranging parameter numbers

Enter the new values and accept these with the **Execute** button.

The following warning is output for an invalid entry.
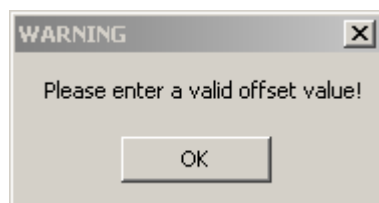
Figure 3-18   Warning message for an invalid entry

Correct your entries.

If your entries were valid, you will find detailed information about moving the parameter numbers in the following **Logs** dialog box on the **Rearrange parameter numbers** tab. In the case of errors, mark the entry and navigate to the problem position with the **Go to** button.
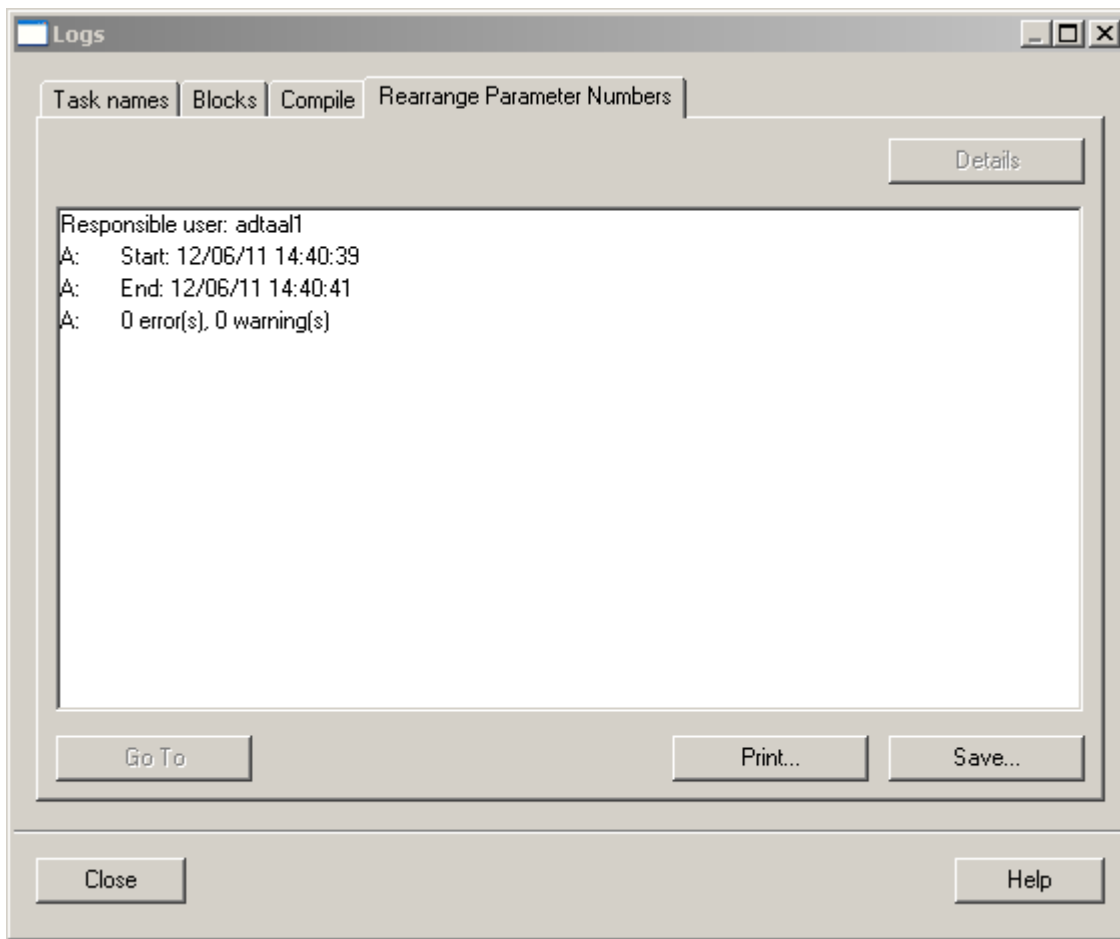
Figure 3-19   Logs dialog box, Rearrange parameter numbers tab

3.5.12   Publishing all connections

In the DCC editor, you can publish the connections of all blocks or the connections of one block.

For further information on the publishing of connections, refer to Creating customer-specific parameters ("declare") (Page 156).

Publishing all connections of all blocks

Publish the connections of all blocks via the menu **Chart -> Publish all connections**.
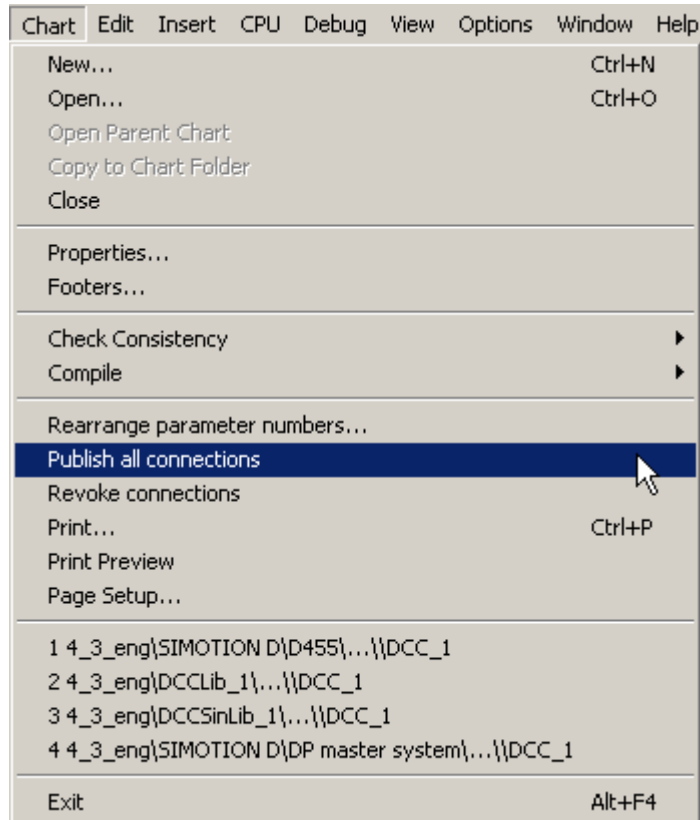
Figure 3-20   Menu **Chart -> Publish all connections**

Publishing all connections of one block

Select a block and publish its connections via the menu **Edit -> Publish all connections**.
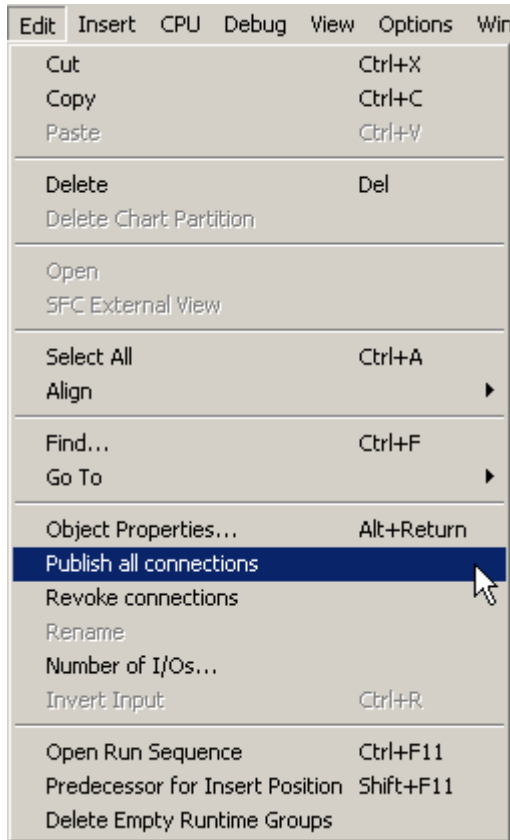
Figure 3-21   Menu **Edit -> Publish all connections**

You can also perform this via the block context menu.
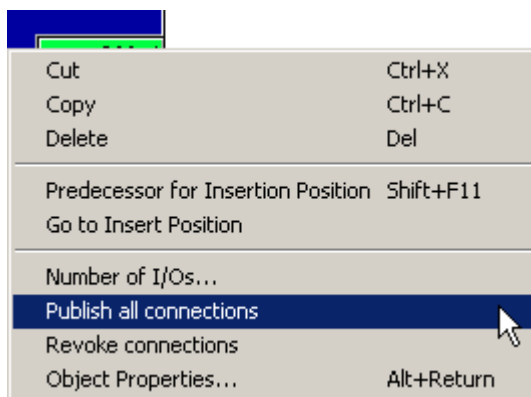
Figure 3-22   Context menu **Publish all connections**

3.5.13   Revoking connections

In the DCC editor, you can revoke the publication of the connections of all blocks and the publication of the connections of one block.

Chart -> Revoke connections

You can revoke the publication of the connections of all blocks via the menu **Chart -> Revoke connections**.
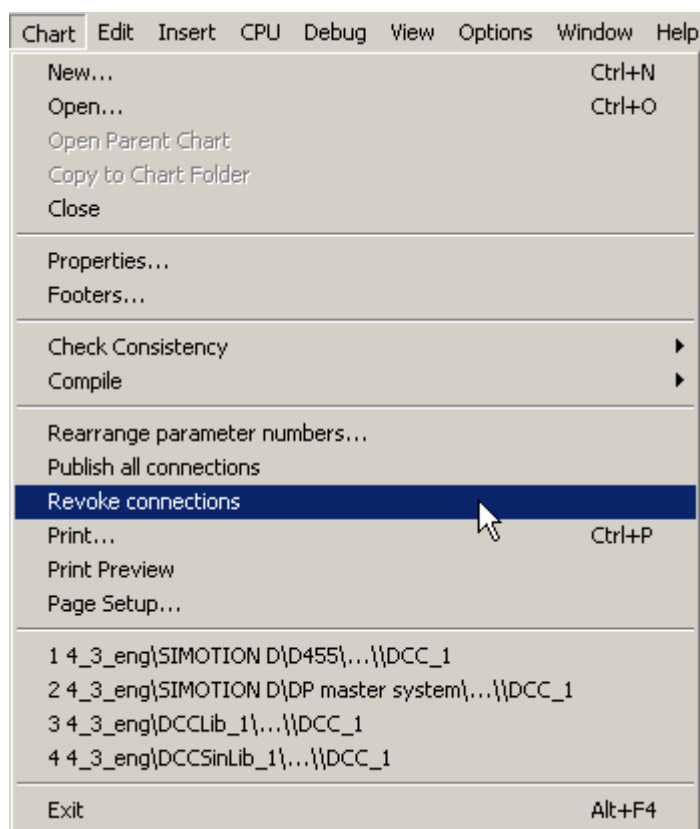
Figure 3-23   Menu **Chart -> Revoke connections**

Edit -> Revoke connections

You can revoke the publication of the connections of one block via the menu **Edit -> Revoke connections**.
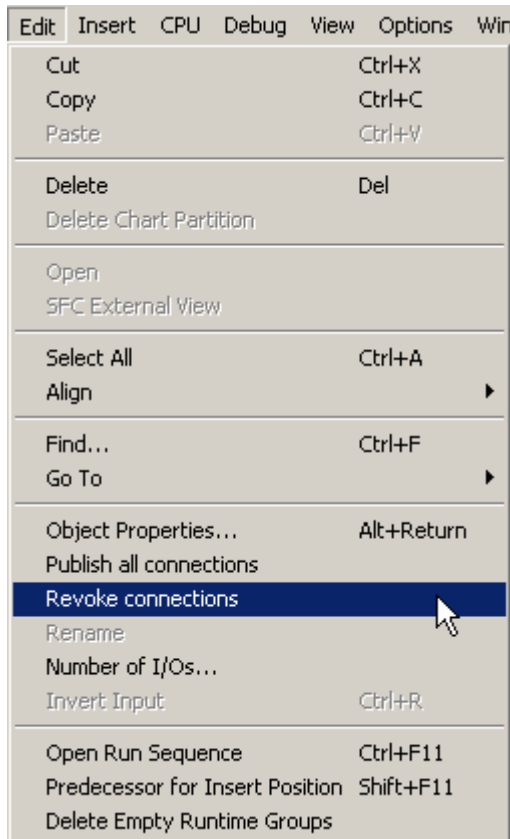
Figure 3-24   Menu **Edit -> Revoke connections**

This function is also available via the block context menu.
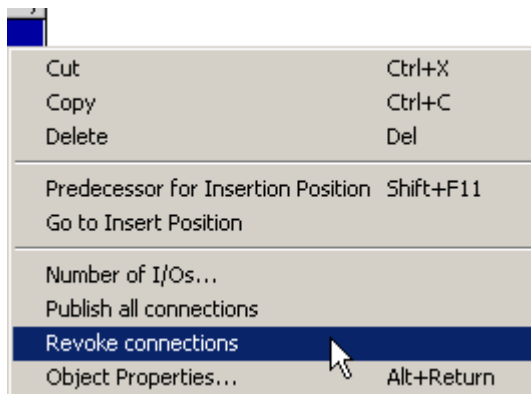
Figure 3-25   Block context menu **Revoke connections**

3.6 Compiling

3.6.1 Consistency check without code generation

General

Some of the conditions that have to be met if a valid configuration is to be created from DCCs can only be checked once charts have been created.

This procedure is carried out automatically at certain points, e.g. when a project or charts are compiled.

Performing a consistency check

The contents of DCCs can be checked at any time.

To check consistency, click **Chart > Check consistency > Charts as program...**

The **Logs** dialog box is displayed automatically after the consistency check. Here, errors are indicated by an "E" and warnings by a "W".

Error log

You can also display the result of the consistency check via **Options > Logs** in the **Consistency check** tab.

Note

If charts are deleted from previously compiled and downloaded projects, this can lead to inconsistencies during the next compilation/download, as the overall chart execution system has to be adjusted.

Note

Cross-chart consistency checks are performed only after the compilation of the charts. The consistency check from the DCC chart always applies to the last compilation result.

3.6.2 Compiling the DCC in the DCC editor

Compiling

Please note that before the first compilation of a chart in a project, the project must first be saved in STARTER/SIMOTION SCOUT (via the menu command **Project** -> **Save** or with the 💾 button).
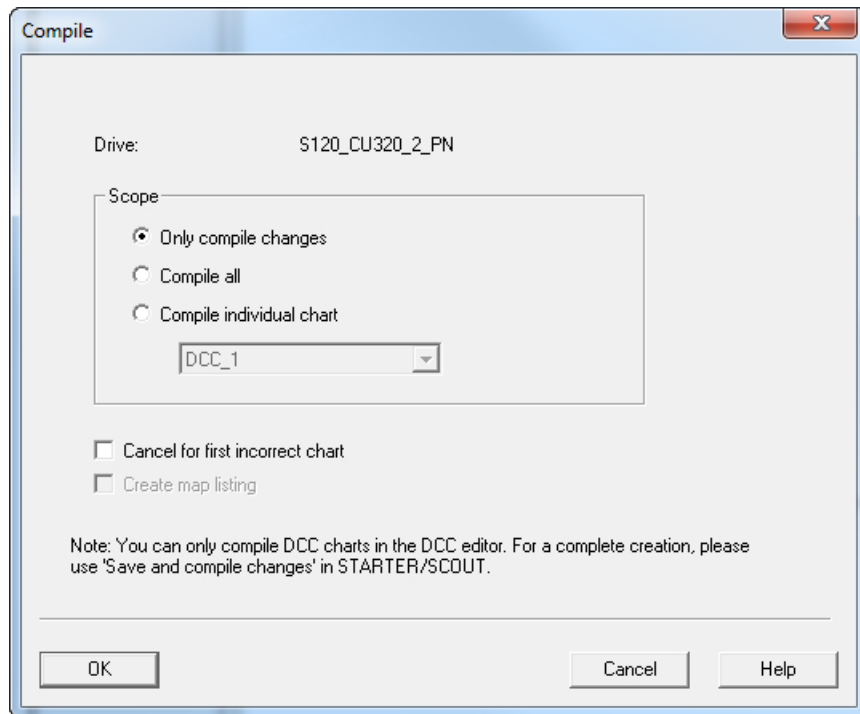
Figure 3-26   Compiling a DCC chart

You can begin compiling with **Chart > Compile > Charts as program ...** or with the 🔘 button.

Compilation options

You can select the scope of the compilation:

- Scope: **Only compile changes**
  Only those parts of the configuration that have been changed since the last compilation are recompiled. When recompiling, this option reduces compilation time.

  Note

  All the configuration charts will be checked for consistency, even if you select the **Only compile changes** option.

- Scope: **Compile all**
  The entire contents of the configuration are compiled, regardless of whether the configuration has been changed since the last compilation.

- Scope: **Compile individual chart**
  The selected chart is compiled, regardless of whether the configuration has been changed since the last compilation.

Additionally, you can select whether a map listing is to be generated and whether it should be canceled in the event of an error occurring in a chart. A map listing is a list of the global objects, the cross reference and the blocks used in the chart.

The map listing is saved in subfolder U7\debug in the project path.

Note

You can only compile DCC charts in the DCC editor. To carry out the creation process in full, please use **Save and compile all** in STARTER/SCOUT.

As of STARTER/SCOUT V 4.2, this function is available under **Save and recompile all**.

As of STARTER/SCOUT V 4.3, this function is available under **Save and recompile all including DCC libraries**.

As of STARTER/SCOUT V 4.4, this function is available under **Save and recompile all.**

Reorganizing OCM variable interfaces

This option is only available for SIMOTION DCC charts. During the reorganization, the OCM variables are removed from the memory image that was already deleted previously. For further information, refer to HMI variables (Page 206).

3.6.3 Error log

On completion of the compilation procedure, a detailed compilation log appears. Here, errors are indicated by an "E" and warnings by a "W".

- To navigate to the block that caused the error, select the error line in the log and click **Go to**, or double-click the error line.

- The log can be displayed again at a later point via the menu **Options > Logs** and, if necessary, can also be printed out.

3.7 Editing configurations further

3.7.1 Editing programs further: overview

This section describes more options for editing an existing program.

The following subjects are covered:

- Changing the block library
- Copy and paste charts or block groups between drive devices

Saving and restoring retain variables - SIMOTION only

DCB block instances may contain retain variables.

Select **Save Variables...** from the context menu of the DCC chart to save the contents of these variables to a file. The values can then be restored from this file. DCC retain data is retained when the platform is changed or the version is upgraded.

3.7.2 Changing the block library

If you want to transfer an existing configuration to a new SCOUT/STARTER version, you can change the library version at a later point. The blocks will be adapted accordingly.

Procedure

1. Open a chart from the relevant configuration and select **Options > Block types** in the menu.

2. The **Import** window is displayed. Click **OK** to close the window. In the window **Import DCB Libraries**, the block libraries installed on your programming device are listed under **Libraries installed in SCOUT/STARTER**. Under **Libraries imported in the chart**, all libraries which have already been imported into this configuration are listed.

3. Check which previously imported libraries have a new version installed, by inspecting the version of the library under **Libraries installed in SCOUT/STARTER** and **Libraries imported in the chart**.

4. Select the library to be updated under **Libraries installed in SCOUT/STARTER** and click **>>**.

5. Click **Accept**.

6. The progress of the update is displayed in a window.

Response

The changes made compared with the old version are transferred to your existing configuration.

3.7.3 Copying of charts or chart sections

In the STARTER and SIMOTION SCOUT engineering systems, charts can be copied within a drive object (SINAMICS) or device (SIMOTION), and between various SIMOTION devices or SINAMICS devices. It is also possible to select block groups within a DCC chart and insert them in other charts of the same device family using copy and paste.

Note

The copying of charts or block groups between SINAMICS devices and SIMOTION devices and vice versa is not supported.

You require a DCC license to copy a DCC chart as a block type into the SINAMICS library.

Copying a chart (SIMOTION)

To copy a SIMOTION DCC chart, proceed as follows:

1. Select the source device from the project view.
2. Open the PROGRAMS subitem of the device.
3. Select an existing DCC chart and select the **Copy** command in the context menu of the chart.
4. Select the target device from the project view.
5. Open the PROGRAMS subitem of the device.
6. Select the **Paste** command in the context menu of the PROGRAMS subitem of the device.

The chart has been copied from the source device to the target device.

Copying a chart (SINAMICS)

To copy a SINAMICS DCC chart, proceed as follows:

1. Select the source drive unit in the project overview.
2. Open, for example, the Control_Unit subitem of the device.
3. Select an existing DCC chart and select the **Copy** command in the context menu of the chart.
4. Select the target drive unit in the project overview.
5. Open, for example, the Control_Unit subitem of the device.
6. Select the **Paste** command in the context menu of the Control_Unit subitem of the drive unit.

The chart has been copied from the source drive unit to the target drive unit.

Note

After copying, you must check whether the interconnections of published block connections in the chart copy have to be adapted to another drive object as a result of the copying. During copying, no automatic adaptation of the interconnections of the chart copy is made to the basic system or to other DCC charts.

DCC SINAMICS: When copying DCC charts, cross-chart interconnections remain in the chart.

DCC SIMOTION: When copying DCC charts, cross-chart interconnections are converted into textual interconnections. They are closed again via **Tools menu -> Close textual interconnections** .

Note

Only one SINAMICS DCC chart per drive object may be created.

Note

When copying a device, the associated charts are also copied.

An XML export or XML import of individual charts is not possible, but the associated DCC charts are contained in the project export.

Note

When copying DCC charts between different devices, the libraries used in the DCC charts are exchanged with the libraries present on the device. In the case of SINAMICS devices, this assumes that a version of the standard library is available for each device. As of SINAMICS 4.4, multiple versions of the standard library are available on a device for compatibility reasons. If the automatic selection option is used, the version with the highest firmware version is used. If multiple library versions are installed for a firmware version, the library with the next-highest version is selected.

For SIMOTION, *dcblibV2_0_simotion4_1_x* is also available on SIMOTION V4.2 devices.

V4.1 libraries are only available on device types that were also available in V4.1. V4.1 libraries cannot be executed on a D455-2 or D455-2. New block types can be found in *dcblibV3_0_simotion4_2*.

Copying of DCC charts to other projects

You can also copy DCC charts from one project to another. As only one project can be opened, you must start STARTER or SCOUT twice for this purpose and start another project.

Note

Please note that you can only copy and paste DCC charts via the engineering system; you cannot cut them.

If you want to copy a SINAMICS DCC chart to another project, then the chart must not use SINAMICS libraries that are only available in the source project. This applies to the libraries in the **SINAMICS LIBRARIES** subitem. In this case, an error message is displayed when the copy action is attempted, and the DCC chart is not copied.

You can use the functionality of the XML export and the XML import in order to always copy DCC charts correctly between projects.

Aborting copying processes with DCC charts

Errors may occur in the following cases when copying DCC charts:

● DCC charts were created using an older version of the DCC editor (CFC version)

● Copying without a DCC license

Copying block groups in the DCC editor

In the DCC editor, you can copy parts from one chart and insert them in another chart. To do this, you must open the source chart and the target chart in the DCC editor.

1. Use the lasso function to select the subsection of the source chart to be copied and select the **Edit > Copy** command in the menu bar.

2. Change to the target chart.

3. Select the **Edit > Paste** command in the menu bar.

The block group has been inserted in the target chart.

Block numbering

When copying blocks or block groups in the DCC editor, the name of the new block is formed as follows: if numeric characters are present at the end of the name, these are all deleted up to the first non-numeric character and replaced by the next free numeric character. It is therefore recommendable to allocate names consisting of letters (and numeric characters) such as block, block1, block2, etc.

Location of the insertion in the execution sequence

Blocks are always inserted into the execution sequence behind the selected block and the most recently added block is always selected automatically.
If a block is to be inserted behind an existing block in the execution sequence, select the block in the overview or in the chart, followed by the function **Predecessor for insert point** in the context menu.
You can change the execution sequence at any time via the **Edit > Execution sequence** command in the menu bar or the ⬛ button.

See also Specifying execution properties (Page 37).

3.7.4   Deleting charts

Deleting a chart in SIMOTION SCOUT

To delete a chart in SIMOTION SCOUT, proceed as follows:

1. Open the SIMOTION SCOUT engineering system.

2. Select the required device in the project overview.

3. Open the PROGRAMS subitem of the device.

4. Select an existing DCC chart and select the **Delete** command in the context menu of the chart.

The chart has been deleted from the device.

Deleting a chart in STARTER

To delete a chart in STARTER, proceed as follows:

1. Open the STARTER engineering system.

2. Select the required drive unit in the project overview.

3. Open, for example, the Control_Unit subitem of the device.

4. Select an existing DCC chart and select the **Delete** command in the context menu of the chart.

The chart has been deleted from the drive unit.

Deleting higher-level elements

DCC charts are also deleted when a higher-level element (e.g. a DO) is deleted.

3.7.5   Search in the project from STARTER/SCOUT

You can use the sheet bar to search for variables and SINAMICS parameters in DCC charts in the open project. The contents of alias definitions can also be found using the search function.

Open the dialog via the menu **Edit > Search in project** or using the shortcut **Ctrl + Shift + F**.

The results are displayed in the search results tab of the detail view.
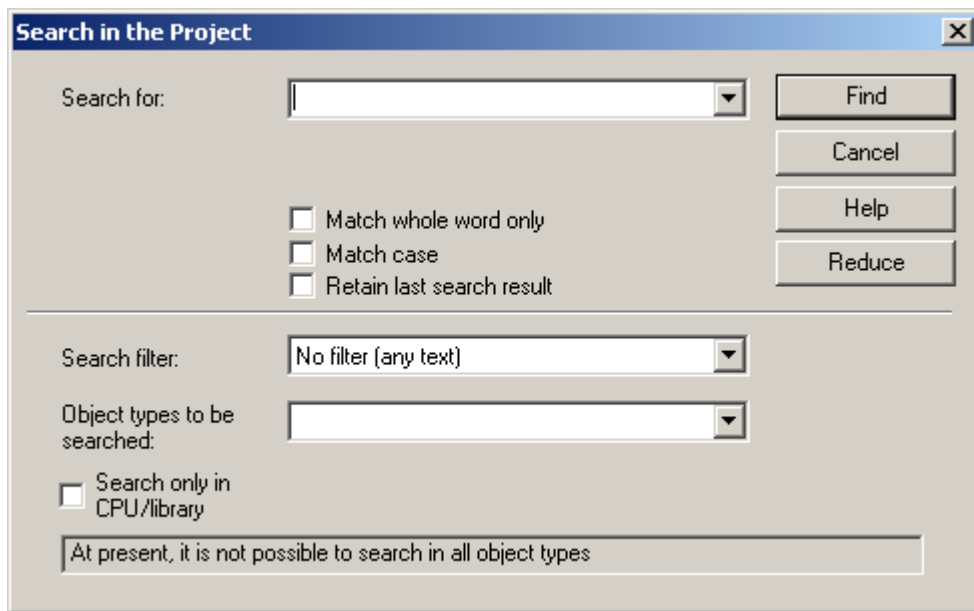
Figure 3-27   Searching in the project

3.7.6   Replacing in the project

The **Replace in project** function is based on the **Search in project** function.

You can use the **Replace in project** function to quickly adjust the interconnections to the system after copying and inserting DCCs.

Open the dialog via the menu **Edit > Replace in project** or using the shortcut **Ctrl + Shift + G**.

When you carry out a replacement, both the results found and the replacement term are displayed in the **Search result** tab of the **detail view**. The text can be edited again here.

Use the **Replace** button to replace all search results selected using the check box.
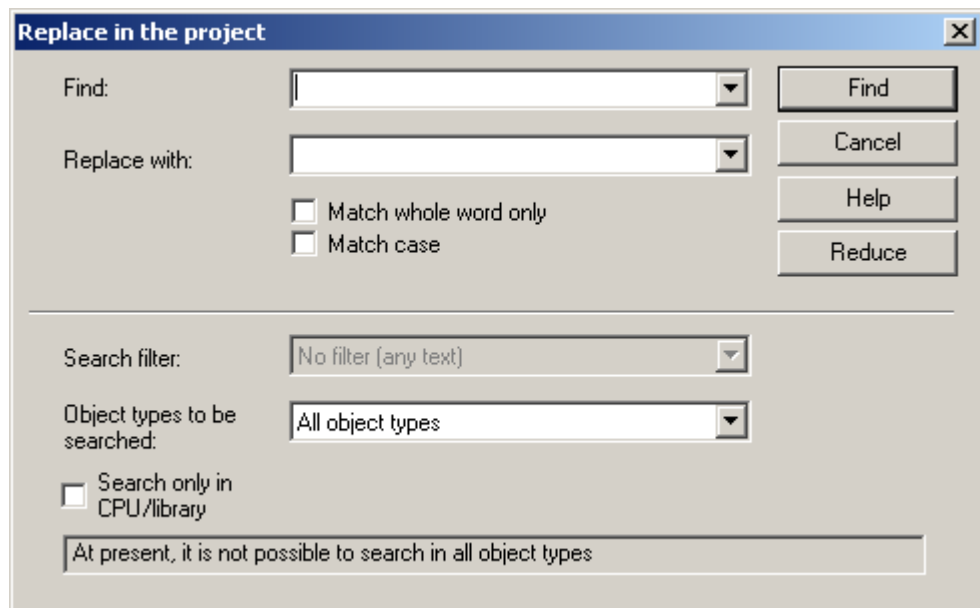
Figure 3-28   Replacing in the project

Note

Symbols that are defined from the DCC, so-called @ parameters, DCC parameters (SINAMICS), links to BICO parameters are not considered when searching/replacing.

Boundary conditions

- As of DCC 2.0.2, the symbols (variable in sheet bar) used in the DCC chart and the contents of alias definitions can also be searched for and replaced.

- The CFC editor must be closed during the search/replace operation.

- If the CFC chart sources have been deleted or the charts have know-how protection, searching/replacing of the sheet bar elements is not possible.

- It is not possible to undo these changes.

3.8 Test mode

3.8.1 Test modes

There are two types of test mode:

- **Process mode**
  Select this test mode if you want to monitor the behavior of individual instances, e.g. for the error analysis. When test mode is activated, all blocks are set to the status **Monitoring off**. In this test mode, you must select the relevant block connections and explicitly log them on for monitoring.

- **Laboratory mode**
  Laboratory mode is used for convenient, efficient testing and commissioning. When test mode is activated, all blocks are set to the status **Monitoring on**.

You can select the desired test mode in edit mode using the menu commands in the **Test** menu. Once a test mode has been selected it is not possible to switch between them.

Test settings

You can specify the **Monitoring cycle** via the menu **Test > Test settings**. This displays the **Test Settings** window.
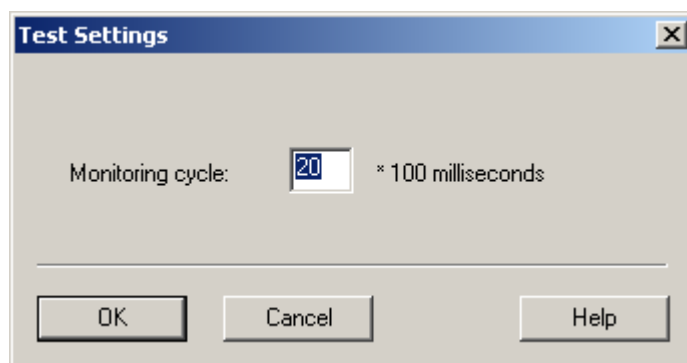
Figure 3-29   Test settings

In this window, you can set the **monitoring cycle** for the objects that have been logged on for testing, i.e. the cycle time for updating these objects. The monitoring cycles can be set in 1 to 100 steps each of 100 milliseconds (SIMOTION). Only whole seconds can be set as monitoring time for SINAMICS.

Note

If you have logged on many objects for the test, it is better to use a slower updating cycle.

3.8.2 Monitoring in laboratory mode

When test mode is activated, the **Monitoring on** function, or ⟨icon⟩, will also be activated for DCC charts in **Laboratory mode**.

This means that, in test mode, you can also display the development of the values for those block connections that are logged on for display, i.e. the values will be read out and displayed cyclically. You can change the options relating to this dynamic value display and the connection parameters in test mode.

Saving settings

The logging on of blocks for monitoring is rejected when the online test is exited. The settings of the block interfaces with regard to the test are saved in the project.

Display of values during the test

Current values of block interfaces are displayed when they are logged on for the test and their block is logged on for monitoring.

3.8.3 Monitoring in process mode

In **process mode**, the **Monitoring off** function, or ⟨icon⟩, is activated. This means that you must first select the blocks that you want to monitor (by highlighting them in the chart) before dynamic display can take place. You then need to execute the menu command **Monitoring on**.

Conversely, you can exclude individual blocks from being monitored if there are too many (highlight the block and execute the menu command **Monitoring off**).

Note

If the monitoring function has been deactivated and you highlight a connection in order to log it on for testing via ⟨icon⟩, monitoring is activated for this connection and for all previously logged-on connections of this block.

By activating test mode, connections are also established with the CPU for all connections listed within a value display window. Activate **monitoring** by selecting the individual connections in the **Monitoring** column.

3.8.4 Logging on/logging off connections for testing

In **edit mode** or **test mode** (process or laboratory mode), you can log on individual block or chart connections for testing:

- Highlight the connection and select **Test > Log on connections** or click the ⟨icon⟩ button in the toolbar.

When this is performed in **test mode**, monitoring is also activated, i.e. the connection is displayed with a yellow background and with its current value. If monitoring has been

deactivated for this block, it will also be activated for all connections that were logged on for testing prior to this.

In **edit mode** or **test mode** (process or laboratory mode), you can log off individual block or chart connections for testing:

- Select **Test > Log off connections** or click the 🔳 button in the toolbar.

Note

Alternatively, it is also possible to log on/log off all connections of a block for testing in edit mode by highlighting a block, selecting **Object properties** via the context menu, and selecting or deselecting the individual connections in the **For testing** column of the **Connections** tab.

3.8.5 Activating/deactivating connection monitoring

You can activate/deactivate monitoring (displaying the current values at the connection) of the block connections logged on for testing:

- Automatically, by activating test mode in laboratory mode.

- Via **Test > Monitoring on** or via the 🔳 button in the toolbar. In both **laboratory mode** and **process mode**, this function concerns only the blocks previously selected in the chart.

- You can deactivate monitoring (meaning that values at the connections will no longer be updated) via **Test > Monitoring off** or by using the 🔳 button. In both **laboratory mode** and **process mode**, this function concerns only the blocks previously selected in the chart.

All input and output values that have been activated for monitoring purposes are updated in accordance with the set monitoring cycle.

3.8.6 Activating test mode

To activate test mode, proceed as follows:

- Click the **Test mode** button in the toolbar
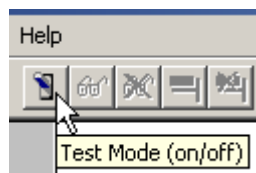
or click **Test > Test mode**.

Test mode is activated. The menu item is identified by a check mark. Depending on the selected test operating mode, the text **Test: RUN(laboratory mode)** or **Test: RUN(process mode)** is

displayed in the status bar with a green background. Any menu functions that are not permitted in test mode are displayed as deactivated (grayed out).

Note

If the DCCs are different in the editor and in the target device, then they can behave differently, see Consistency of charts in test mode (Page 73).

Requirements for starting test mode

The use of test mode requires an online connection to the device.

As long as the DCCs are identical online and offline, test mode is activated immediately. If the DCCs do not have the same version in the RT system and in the engineering system, values can still be monitored in test mode, but in order to make further changes the consistency must first be restored by uploading the changes from the target system (in this case the changes are not visible in the editor) or by downloading the current version from the engineering system.

Note

Once the project has been uploaded from the target device, it needs to be saved before DCC test mode can be started.

If you have made changes to connections or block types in the DCC, you will be informed that you can still monitor and trace values when test mode starts.
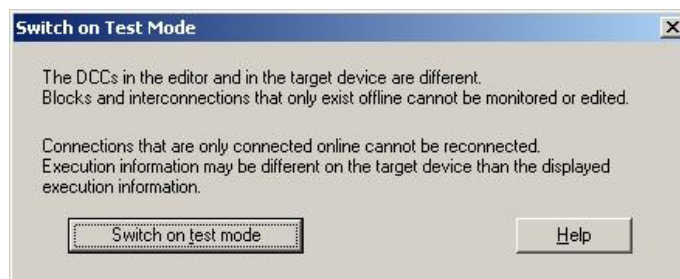
Figure 3-30   Test mode - changes to connections or block types

Online changes can only be performed after the DCC has been recompiled and downloaded to the target system. You are informed about this fact when starting test mode and can return to editing mode or activate test mode for monitoring purposes - as shown in the following dialog box.
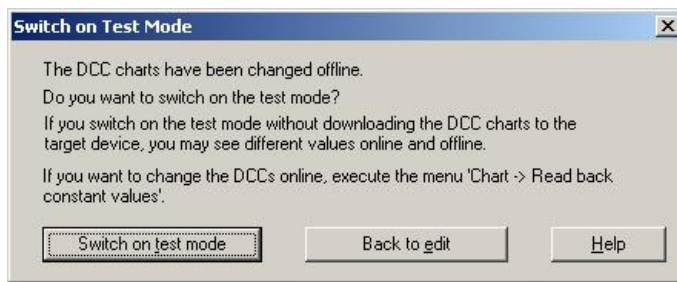
Figure 3-31   Test mode - loading changes to the target device

3.8.7   Monitoring test mode

In test mode, the values of the connections that have been logged on for monitoring are displayed with a yellow background.

Logging on connections for testing

You must explicitly log on to test the connections to be tested. For SIMOTION, the log on to the test can be performed either using the **Log on connection** button in the menu bar or using the context menu of the block connection. Select the **Log on connection** item in the context menu of the connection. For SINAMICS, go to the **Object properties** of the block connection (right-click) and activate the **For test** property. You can also log on the connections directly for the test by clicking with pressed Ctrl key.

Changing input values

You can also change all values of non-interconnected inputs in test mode. To show how changing a value affects execution, assign a new value to an input as follows:

● Double-click the input to be monitored.

● The **Properties - Connection** window appears. Enter the new value for the block input and confirm with **OK**.

● In the chart, you can now see how the value changes on the associated output.

> **Note**
>
> The changed value is only visible when the connection has been logged on for testing.

Logging off connections from testing

You can log off connections logged on for testing. The log off from the test can be performed either using the **Log off connection** button in the menu bar or using the context menu of the block connection. Select the **Log off connection** item in the context menu of the connection. You can also log off the connections directly from the test by clicking with pressed Ctrl key.

3.8.8 Enabling the value and trend display during a test

In test mode, you can use the value and trend display to analyze the input and output values of blocks.

Note

The dynamic display is limited to 256 values.

It includes the structure of a block connection with more than 256 single elements. As such, you can only insert individual elements selectively into the dynamic display and not the block connector in its entirety.

Drag individual elements to the dynamic display by means of drag-and-drop to display currently relevant values.

Please note that monitoring values in the dynamic display influences the performance of the target device

---

Enabling the value and trend display

You can open the value and trend display using the **View > Value display** and **View > Trend display** commands in the menu bar of the DCC editor. Each of these is an autonomous program window. The windows can be arranged using the minimize and maximize functions.

The block interfaces can now be added to the value or trend display using the **Insert in value display** and **Insert in trend display** commands in the context menu. Current values are displayed when the test mode is switched on.
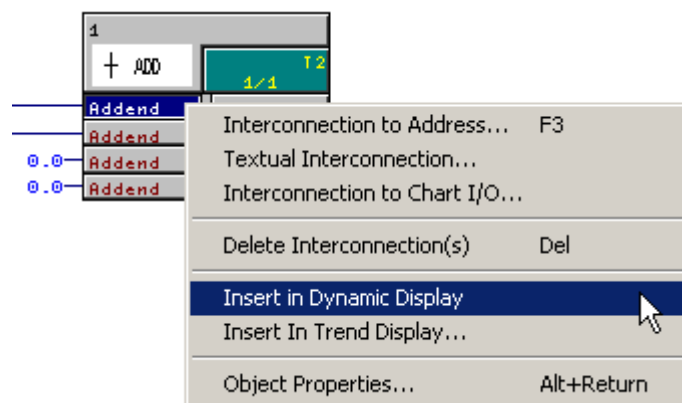
Figure 3-32   Inserting a block pin into the value or trend display

Settings in the Trend Display window

The **Trend Display** window displays the block connections added to the trend display in the form of curves. Each inserted block connection is called a **channel**, whereby a **lower limit** and an **upper limit** can be specified for each channel.

The number of sample values to be displayed on the time axis is specified in the **Display** area of the **Trend Display** window.

The desired trace parameters can be set in the **Trace Parameters** window, which can be accessed via the **Change** button.

3.8.9 Editing DCCs in test mode

To a large extent, you can continue to edit your configurations during test mode. An overview of the changes that can be made during test mode is shown below:

Table 3-3   Further editing of configurations during test mode

Handling blocks (Page 34)	
Inserting blocks	Call the block catalog using the **View > Catalog** command. Open the block family and use drag-and-drop to move the selected block to the working area.
Deleting blocks	Select the block and use the **Edit > Delete** command to remove it. In DCC SINAMICS, blocks for which at least one connection has been published as a parameter cannot be deleted. The SAV, SAV_BY, SAV_D, SAV_I and STM blocks cannot be deleted or inserted online.
Moving blocks	Select a block and drag it to the desired position in the chart.
Renaming blocks	Select the block, right-click and select the **Rename** command. • The new block names apply only to the accesses to the input/outputs of the block using Java or Trace; the old names remain valid here. • For those blocks that use retain data, the retain data block on the device is also assigned the new name.
Interconnecting blocks (Page 40)	
Creating interconnections	In the DCC, select the block connections between which you want to establish a connection. Note that in DCC SINAMICS, block inputs published as BICO parameters can only be connected to block outputs published as BICO parameters or BICO sources of the basic system. BICO inputs and BICO outputs of the basic system can still only be connected to block outputs and inputs published as BICO parameters in the DCC.
Deleting interconnections	In the DCC, select the block connection that you want to disconnect. Then select **Edit > Delete** to delete it.
Moving interconnections	In the DCC chart, select the block connection that you want to move. Then move it using drag-and-drop. When moving BICO interconnections, the signal can acquire the value 0 for a few cycles. The no longer interconnected input on the original connection is permanently assigned the value 0.
Changing the signal value of an input	Double-click the block input for which the value is to be changed. The "Properties - Connection" dialog box appears in which you can change the value.
Deleting global interconnections to the sheet bar	In the DCC, select the block connection that you want to disconnect. Then select **Edit > Delete** to delete it.
Comments (text)	

Inserting comments (text) in the chart	Select the **New text** command via the directories of the block types and insert it in the chart using drag-and-drop. Release the mouse button at the required point.
Changing comments (text) in the chart	You can move the text field in the DCC by selecting it and then dragging it to the desired position. You can change your comments by double-clicking the text field and then entering or editing your text.

3.8.10 Deactivating test mode

Deactivate test mode to return to edit mode.

● Click the **Test mode** button in the toolbar.
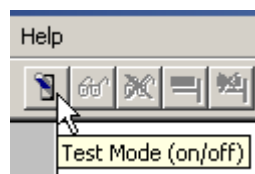
or click **Test > Test mode**.

Edit mode is reactivated.

3.8.11 Changing online during test mode

3.8.11.1 Preliminary remark

As a general rule, test mode is used for making online changes to values, interconnections, and block instances without stopping the system.

It is not necessary to recompile after changes have been made online.

The changes are made in the target device and offline data storage area at the same time.

However, they must be saved with **Copy RAM to ROM** before Power Off , otherwise they are lost. This is especially important with regard to SAV blocks.

Note

Please note that online changes can only be made in test mode.

In test mode, the "Server busy" message may be intermittently displayed if you attempt to change values online.

You may need to acknowledge this message several times before the value that is being changed online is accepted.

Repair the project using an XML export/import.

Note

In SIMOTION SCOUT, deactivate the option **Options → Settings → Save → Automatic back-up copy of the project data** when working with large projects to prevent a drop in performance.

Otherwise, the entire project would be saved in test mode each time a change was made.

3.8.11.2 Changing values at block inputs online

Requirement

Only signal values at block inputs that are not interconnected can be changed online.

Procedure

The **Properties** - **Connection** window is opened by double-clicking the desired block connection. A new numerical value can now be entered for the block input in the **Value** line. The new value takes effect and is displayed in the chart when **Accept** is clicked. The window is closed by clicking **OK**.

Note

DCC-SINAMICS 2.5: Only signal values of block inputs can be changed online that have not been published as BICO parameters.

As of SINAMICS 2.6., all non-interconnected block inputs can be changed in test mode. They do not have to be published.

Note

DCC SIMOTION: The signal value at the block input can be changed online, but with active execution group is overwritten in the next cycle. Whether the signal value at the block input can be changed online does not depend on whether it has been declared as an HMI variable or not.

3.8.11.3 Deleting an interconnection online

Procedure

In the DCC, select the block connection that you want to disconnect. Then remove this with **Edit > Delete** or with the Del key.

Result

The connecting line between the connections disappears and the last value that was transferred on the connection appears as input value at the connection.

Note

DCC SIMOTION: Interconnections to chart connections cannot be deleted online!

3.8.11.4 Establishing an interconnection online

Procedure

In the DCC chart, select the block connection from which you want to establish an interconnection and drag an interconnection to the block connection to which the interconnection is to be established.

Result

The connecting line between the selected connections is established and the current value that has just been transferred appears at the output.

Note

Up to and including DCC-SINAMICS 2.5, a block input published as a BICO parameter may only be interconnected with outputs published as BICO parameters or with BICO outputs of the basic system.

Block connections cannot be published in test mode, i.e. new @ parameters inserted.

Note

DCC SIMOTION: Connections to published block inputs in the DCC or global operands cannot be recreated online.

3.8.11.5 Moving interconnections online

Procedure

Select the required interconnection and move it with drag&drop.

Note

When moving BICO interconnections, the signal can acquire the value 0 for a few cycles.

The no longer interconnected input on the original connection is permanently assigned the value 0.

3.8.11.6 Inserting a block online

Procedure

Call the block catalog using the **View > Catalog** command. Open the block family and use drag-and-drop to move the selected block to the working area.

The block instance is calculated with the next cycle.

The inserted block instance is assigned a standard name that can subsequently be changed online: Select the block, right-click and select **Rename**. Recompilation and loading is only required if the trend display, trace or monitoring of block connections has been activated.

Note

Whether a block can be added or deleted online is described under "Configuration data" in the online help for blocks.

Note

Renaming blocks

When blocks are renamed in online mode, a new block is created that replaces the existing block. This is then recalculated. The initialization values are newly set for the block.

Online renaming allows meaningful names to be assigned also for the online inserting of module instances.

3.8.11.7 Deleting a block online

Procedure

First delete all output interconnections by selecting the connection and then selecting **Edit > Delete** in the editor menu or with the Del key. Then select the block and remove it with the Del key or the **Edit > Delete** command.

Note

In SINAMICS, blocks for which at least one connection has been published as a parameter cannot be deleted.

Note

DCC SIMOTION: Blocks with interconnections to chart connections cannot be deleted online.

Note

The STM block cannot be deleted or inserted online.

3.8.11.8 Inserting comments in the chart

Procedure

Select the **New text** command via the directories of the block types and insert it in the chart using drag-and-drop. Release the mouse button at the required point.

3.8.11.9 Changing comments in the chart

Procedure

You can change your comments by clicking the text field and then entering or editing your text.

You can adapt the size of the text field to your requirements by selecting the black points at the corners and the sides and dragging them to the desired size.

3.8.11.10 Moving blocks in the chart

Procedure

Select the block and drag it to a suitable free position in the chart.

3.8.11.11 Consistency of the charts in test mode

As long as the DCC charts are identical in the DCC editor and in the target system, DCC charts can be monitored and changed in test mode.

If the DCC charts are different in the engineering system and in the target system, changes can still be made online as long as the DCC charts are not compiled.

If the charts are different in the editor and the target device and the charts have not yet been compiled, the following dialog box appears:
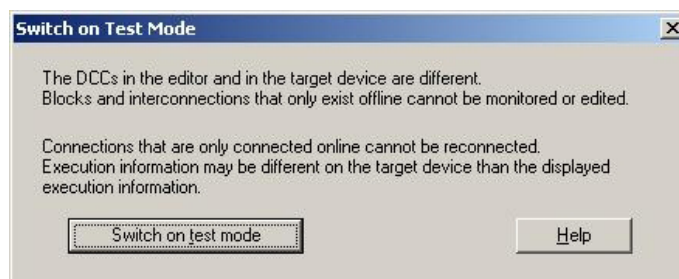
Figure 3-33   Activating test mode with inconsistencies

Click "Switch on test mode" to confirm the dialog.

If only the constant values of the chart are different in the CFC editor and in the target system, the user can recompile the DCC chart and transfer the changes from the target system to the CFC editor with **Options > Read back constant values**. Monitoring, tracing and further online changes can then be performed in test mode.

If the DCC charts differ online and offline and the DCC charts have already been compiled, a prompt appears that the DCC charts are different online and offline.

● If only the constant values of the charts are different, you can restore the consistency using the menu command **Options > Read back constant values**.
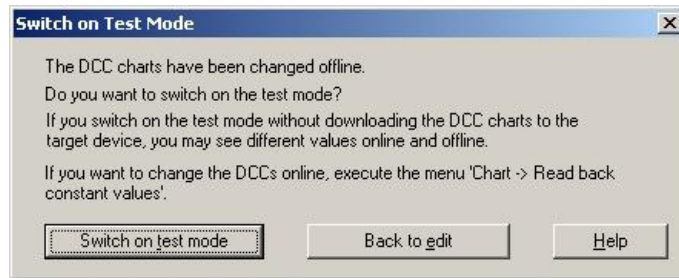
Figure 3-34   Activate test mode - Read back constant values

● If the DCC charts differ online and offline and have already been compiled, a prompt appears that the DCC charts are different online and offline. Download the changes to the target system so that monitoring and changes can be performed in test mode.
The download is performed when the target device is in the STOP state. With SINAMICS, the download only functions in the "Power-on inhibit" and "Ready for power on" operating modes.
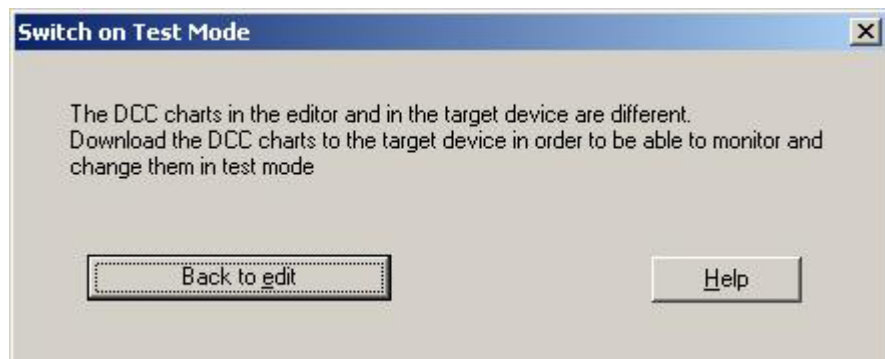
Figure 3-35   Activating test mode

3.9 Reference data

3.9.1 Chart reference data

As well as displaying the following information graphically in the DCC, you can use the "Chart reference data" (**Options > Chart reference data**) function to display it in the form of a list and print it out:

- Cross references of accesses to different objects

- Execution sequence
  Graphic representation of the entire run sequence of a CPU.

You can use this to check your configuration structure.

You can display and print out the following lists of chart reference data:

- Operand cross references
  This list displays all global operands used on the CPU along with the elements that access them.

- Execution group cross references
  This list displays the existing accesses of any DCCs for all execution groups.

- Block type cross references
  This list displays the block types used and the positions (on the DCC) where they are used.

Note

You can also generate the reference data of a DCC or a DCC library via the context menu with the **Reference data > Generate** command and then display it via **Reference data > Display**.

You can then always display the reference data again, but you only have to generate it once when first called or after changes to the @ variables.

3.9.2 List of block types

The list of block types shows where they are used. Unused block types can be deleted.

You can display the list by opening the **Chart Ref: Display Chart Reference Data** window via the **Options > Chart reference data** menu item. Now select the **View > Block types** menu item or use the ⊞ button in this window.

Column heading	Meaning
Block type	Block name (e.g. ADD)
Chart	Name of the DCC in which the block is used
Block	Name of the block instance (e.g. Integrator1)
Block comment	Block comment

Note

Double-clicking one of the displayed block types opens the associated chart and the selected block type is highlighted.

3.9.3 Cross References List Execution Groups (only for DCC-SIMOTION)

The list shows the existing accesses from arbitrary DCCs to the enable attribute of the execution group for all execution groups of the active CPU (the enable attribute can be used to switch individual execution groups on and off, see Enable attribute, execution groups (Page 226)).

You can display the list by opening the **Chart Ref: Display Chart Reference Data** window via the **Options > Chart reference data** menu item. Now select the **View > Chart element cross references > Execution group** menu item or use the ⬚ button in this window.

Column heading	Meaning
Execution group	Group name
(R/W)	Read (R) or write (W) access
Chart	DCC name
Chart element	Name of the interconnected block and connection
Element comment	Block comment
Type	Block type (e.g. ADD)

Note

The names selected in the execution system (i.e. the names of the execution groups and execution levels) for each device must be unique.

3.9.4 List of operand cross references

This is a list of the global operands used on the CPU or the drive unit, along with the block connections that they are connected to.

You can display this list when you display the **Plan Ref: Chart Reference Data** window from the **Options > Open chart reference data** menu item. Now select the **View > Chart element cross references > Operand** menu item or use the ⬚ button in this window.

Table 3-4   Global operands

Column heading	Meaning
Symbol	Name of the global operand
Address	Blank

Column heading	Meaning
Data type	Keyword of the data type used
(R/W)	Read (R) or write (W) access
Chart	DCC name
Chart element	Name of the interconnected block and connection
Element comment	Block comment
Type	Block type (e.g. ADD)

3.10 Library handling

3.10.1 Library compatibility

Compatibility

Please refer to the following table for information on which kernel version supports which DCB libraries.

Kernel version	DCB lib version
SIMOTION 4.1.5	4.1.2, 4.1.4, 4.1.5
SIMOTION 4.2	4.1.5, 4.2
SIMOTION 4.3	4.1.5, 4.2, 4.3
SINAMICS V2.5.SP1	SINAMICS V2.5
SINAMICS V2.6.x	SINAMICS V2.6
SINAMICS V4.3	SINAMICS V4.3
SINAMICS V4.4	SINAMICS V4.3, SINAMICS V4.4
SINAMICS V4.5	SINAMICS V4.3, SINAMICS V4.4, SINAMICS V4.5

3.10.2 Handling DCC libraries and block types

The creation of SINAMICS DCC libraries is possible from DCC 2.1.

You can exchange block types within the DCC chart using the **Options -> Block types** menu command. If there are no DCC chart sources, the versions of the basic DCB libraries on the chart can be exchanged by selecting **Block types** from the context menu. This enables you to use the same DCC library on different device versions. The relevant DCC license is required for this.

Changes to execution groups are made automatically when saving under a different target device.

If a chart containing multiple execution groups is copied from the device to the library, error messages will appear during compilation. The user needs to correct the execution groups in this case.

It is possible that libraries and block types with the same names are present in the separate directories for the SIMOTION and SINAMICS libraries. Block type names must be unique within a library.

Note

Use the STM block in DCC library blocks that are only instantiated once per DO.

3.10.3   Importing block libraries

When first creating a chart in SCOUT/STARTER, you may be prompted to import a block library for the selected device platform.

The import process involves the DCB library that is already installed in SCOUT/STARTER being mapped to the Step7 data storage area for the device or DCC library, thus making it available for the charts in the DCC/CFC editor. DCC libraries located in the same project can also be imported for SIMOTION and SINAMICS devices.

You need to install libraries in SINAMICS/SIMOTION and then import them for the chart containers (**Programs** folder) before you can use the block types contained in DCC charts.

Only libraries that are appropriate for the device may be used. Only one version of a library should be imported. For a SIMOTION device 4.1, only libraries as of Version 4.1 + ServicePackX (e.g. 4.1.0, 4.1.5) should be imported.

Note

You can no longer use a SINAMICS 2.6 library as of SINAMICS Version 4.x.

If the DCC library required in the chart is not available for selection, this means that a basic library used within the library is not valid for the current device. You can change the basic library in the DCC editor under **Options > Block types**. If a chart source is not available, you can call up the **Block types** menu command from the context menu in the DCC library.

Name ambiguity in blocks from different libraries

If two blocks from different libraries (in the case of SIMOTION) have the same name, then the block from the library whose name comes first in the alphabet will apply.

If a DCC chart has not been inserted for a chart container (Device/Library), it is not possible to import a library.

Note

For SINAMICS, the libraries must always be imported under the device and not under the drive object.

Deleting the last chart of a device clears the selection of the DCB libraries.

When importing a DCC library, it must be compiled.

How to import block libraries:

- Open a chart from the relevant configuration and click **Options > Block types**.
  The **Import** window appears. Click **OK** to close the window. In the **Import DCB Libraries** window, the block libraries installed on your programming device are listed under **Libraries available for the device**. Under **Libraries imported in the chart**, all libraries which have already been imported into this configuration are listed.

- Select the library to be imported under **Libraries available for the device** and click **>>**. The import process is triggered when you click **Accept**.

Checks are performed when a block library is imported.

The names of the block libraries are defined on the basis of a naming convention. The individual parts of the name are explained in the following table. The part in question is shown in bold each time.

Table 3-5　　　Naming convention for block libraries

Part of the block library name	Meaning
TPdcblib_SIMOTION_4_1_2.2.0 [7.0]	Library identifier
TPdcblib_**SIMOTION**_4_1_2.2.0 [7.0]	Target platform
TPdcblib_SIMOTION_**4_1_2**.2.0 [7.0]	Target system version
TPdcblib_SIMOTION_4_1_2.**2.0** [7.0]	Library version
TPdcblib_SIMOTION_4_1_2.2.0 **[7.0]**	Build version

3.10.4　　　Updating the block library

When a DCC chart is open, you can update the block libraries via the menu **Options > Block types**.

For DCC libraries without DCC chart sources, the versions of the basic DCB libraries can be exchanged by selecting **Block libraries** from the context menu.

Note

The library concerned must be selected for this purpose.

How to update block libraries:

- Open a chart from the relevant configuration and click **Options > Block types**.

- The **Import** window is displayed. Click **OK** to close the window. In the window **Import DCB Libraries**, the block libraries installed on your programming device are listed under **Libraries installed in SCOUT/STARTER**. Under **Libraries imported in the chart**, all libraries which have already been imported into this configuration are listed.
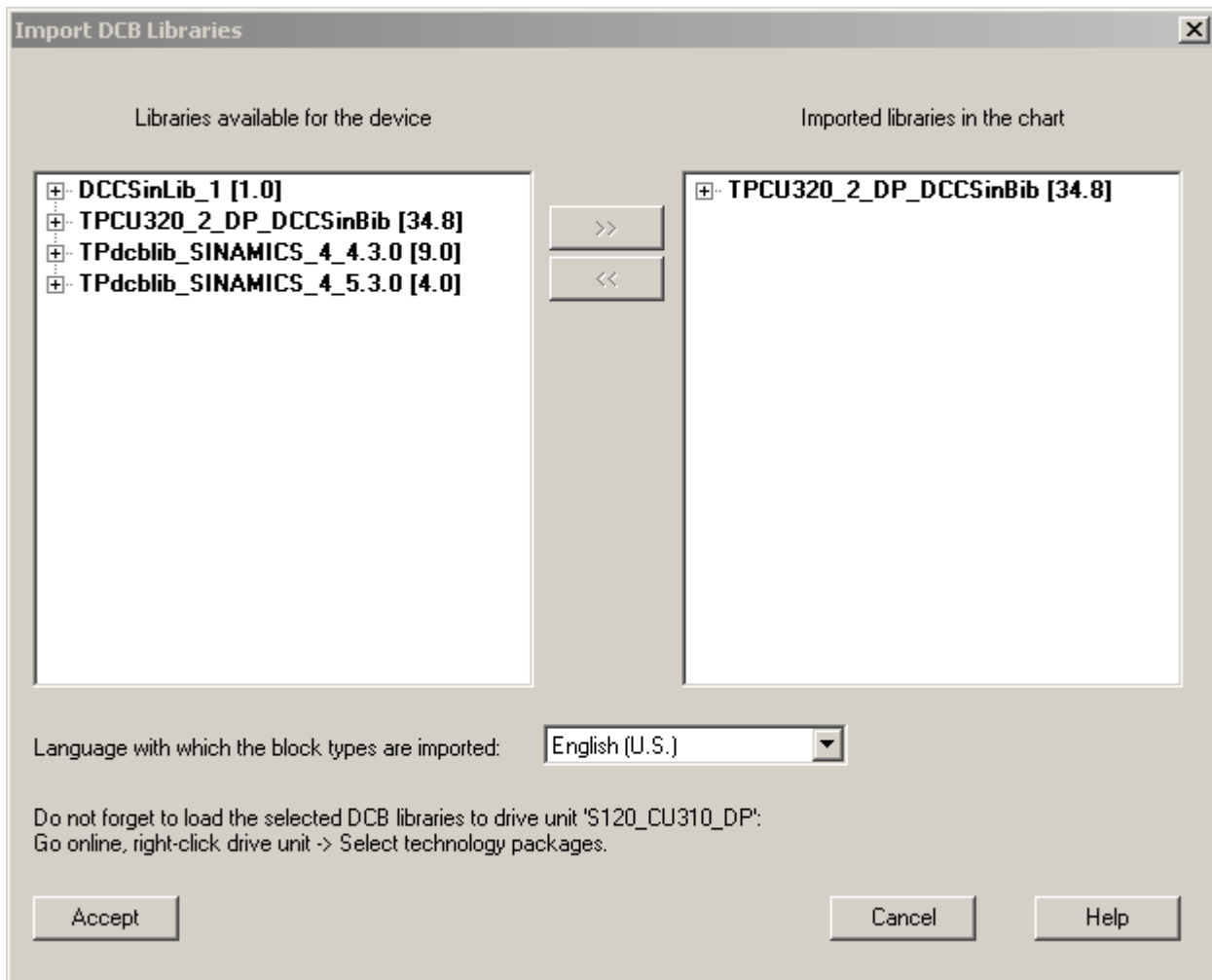
Figure 3-36   Updating the block library

Updating libraries after a device upgrade

As of STARTER 4.2, the system automatically swaps in the correct library version after a device version has been upgraded.

Detailed information about upgrading a device version can be found in the section titled Software upgrade and module exchange (Page 120).

Note

The standard library is automatically installed during setup, but if you do uninstall it accidentally, you can reinstall it by means of the DCC setup process.

The libraries for SINAMICS can also be installed via the SSPs (SINAMICS Support Packages).

If you wish to reinstall the DCB library from the SSP DVD, switch to directory CD_1\SSP\Disk1 and unzip file dcblib.zip into any directory. Install the library dclbib_Vx.y_sinamics_w.z contained in unzipped files.

As of DCC V2.1 (STARTER 4.2), the libraries for SINAMICS are also available via the menu command **Options -> Install libraries**. See also Installing and uninstalling DCB libraries (SINAMICS).

The libraries for SIMOTION are also available via the menu command **Extras -> Install libraries**. See also Installing and uninstalling DCB libraries (SIMOTION) (Page 102). Here you will also find information on where installable DCBLIB standard libraries are located and how to reinstall these.

- Check which previously imported libraries have a new version installed, by inspecting the version of the library under **Libraries installed in SCOUT/STARTER** and **Libraries imported in the chart**.

- Select the library to be updated under **Libraries installed in SCOUT/STARTER** and click **>>**.

- Click **Accept**.

- The progress of the update is displayed in a window.

Note

In SINAMICS, you can check a selected device in the engineering system using the **Select technology packages** function in the context menu as to whether all required block libraries and technology packages have been activated.

Note

In SIMOTION, the library is automatically loaded to the device during the project download. In SINAMICS, this action must be performed explicitly by the user before the project download. (Page 182)

The library to be imported is checked to see whether it contains a block type that has already been imported with another library.

Error log

If errors are detected when a library is being imported, an error log with details of the causes of the errors is displayed.

3.10.5 Exchanging the basic library version for installed libraries

The basic libraries are selected from the DCC charts used in the dialog **Options → Block types** of the DCC editor.

If there are no DCC sources available at the DCC charts, you can open a dialog via the context menu **Block types** of the DCC chart, where you can exchange the block libraries used.

Libraries installed in SCOUT	Imported libraries in the chart
TPdcblib_SIMOTION_4_3.4.0 [111.0]	TPdcblib_SIMOTION_4_3.4.0 [111.0]
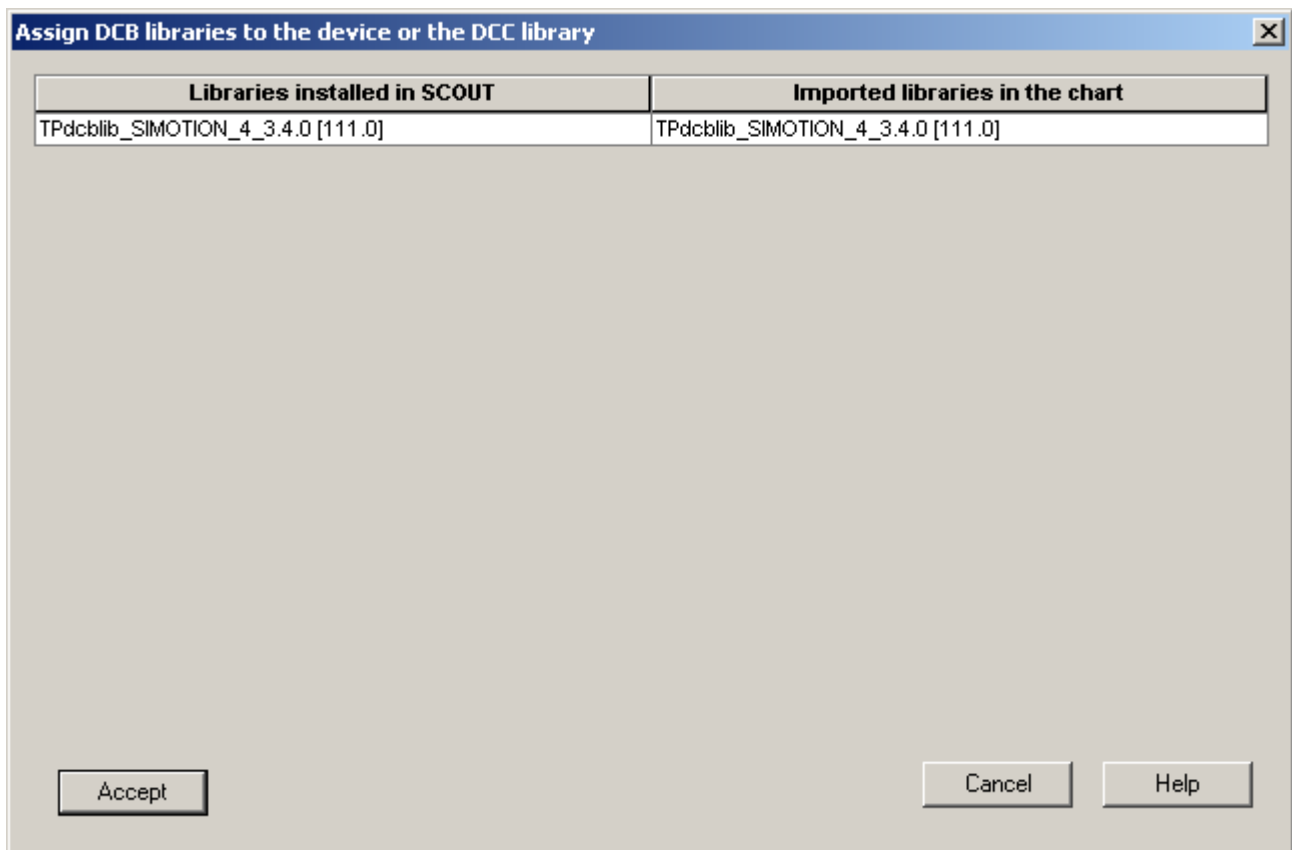
Figure 3-37   Block types

Mark the library in the left-hand column and press **Accept.**

Up to DCC V2.0.1, the version of the basic library used is permanently defined in a DCC library.

Exactly the same basic library version that was used to create the DCC library must be installed.

If the library sources have been deleted, it is not possible to select a new version of the basic library in the typical library.

With DCC 2.0.2, you can combine an installable library that has been supplied with different versions of a DCB library that is being used, provided the interface for the blocks called does not change.

3.10.6 Changing the block library language

You can customize the language of the block type comments via the menu **Options > Block types**.

How to change the language of the block types in a block library:

Checks are performed when a block library is updated.

● Open a chart from the relevant configuration and click **Options > Block types**.

● The **Import** window appears. Click **OK** to close the window.
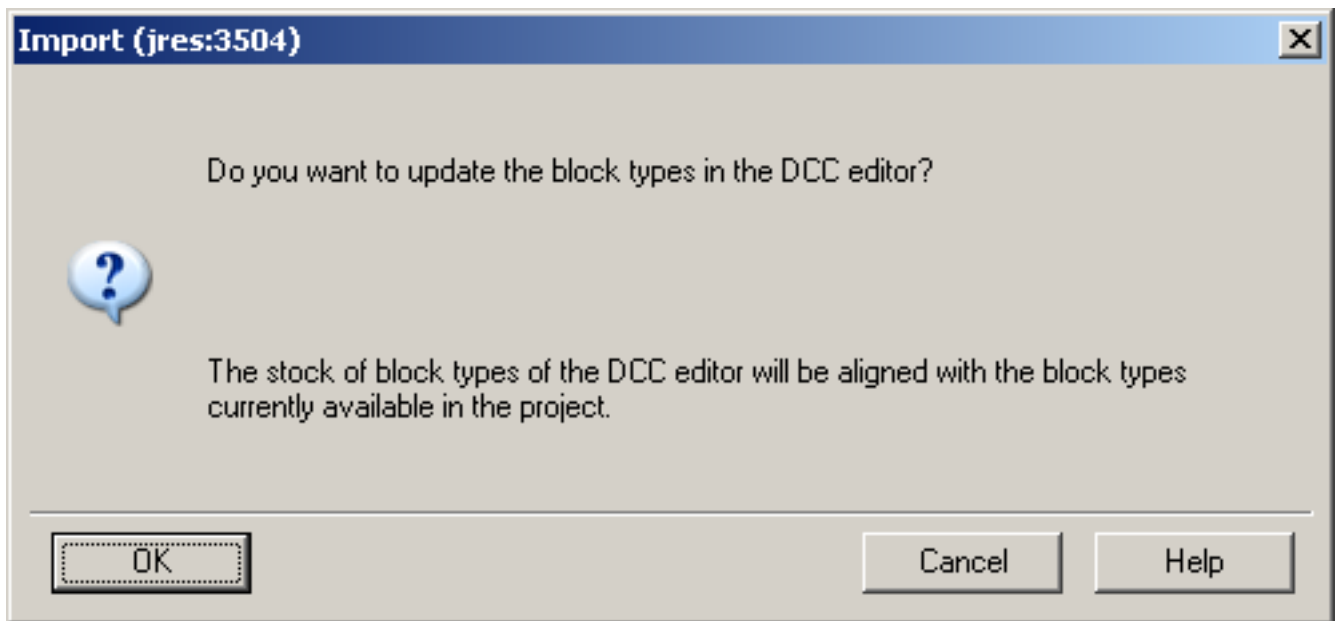
Figure 3-38   Import window

● The **Import DCB Libraries** window appears. The block libraries installed on your programming device are listed under **Libraries installed in SCOUT/STARTER**. Under **Libraries imported in the chart** all libraries which have already been imported into this configuration are listed.

● Select the desired language in the selection list of available languages. The selected language affects all imported libraries.
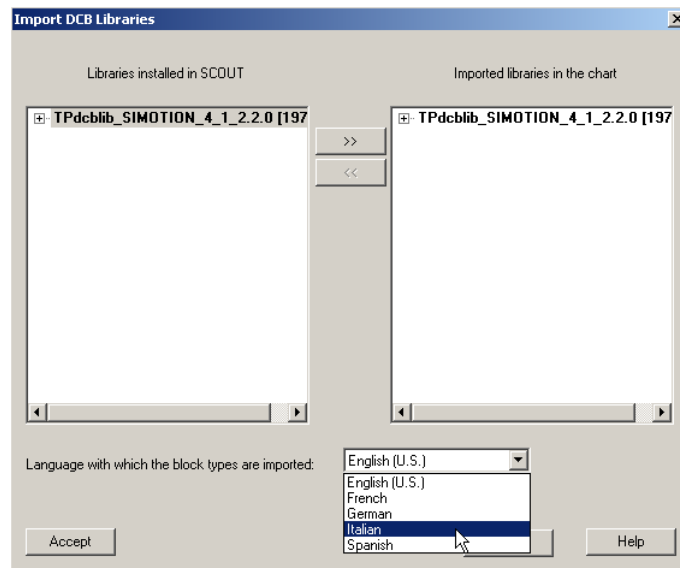
Figure 3-39   Changing the block library language

● Click **Accept**.

● The progress of the update is displayed in a window.

Note

Comments regarding the block types remain in the language in which they were created. The comments regarding block pins are translated if they have not already been altered manually.

3.10.7   Removing block libraries from the configuration

You can delete block libraries whose block types are no longer required in the configuration concerned via the menu **Options > Block types**.

How to delete block libraries:

● Ensure that the block types from the block library to be deleted are no longer to be used at all in the charts of the configuration.

● Open a chart from the relevant configuration and click **Options > Block types**.

● The **Import** window appears. Click **OK** to close the window.

- The **Import DCB libraries** window appears. The block libraries installed on your programming device are listed under **Libraries installed in SCOUT/STARTER**. Under **Libraries imported in the chart**, all libraries which have already been imported into this configuration are listed.

- Select the library to be deleted under **Libraries imported in the chart** and click **<<**. The deletion process is triggered when you click **Accept**.

Checks are performed when a block library is deleted. All unused block types of the library will be removed from the selection list of the block types in the DCC editor; used block types remain in the list. The chart cannot be compiled without an imported library if instances of the library are still present in the chart.

3.11 Creating block libraries

3.11.1 Fundamentals

The DCC editor offers a function for saving a chart as a **block library (DCB library = typical)**. This kind of chart relates to a connection created by the user, which features an interface so that it can be reused and is saved in a library. The charts as block library (DCB library) function therefore include the option of know-how protection, since the configured DCCs are only available as a transparent block within a block library. The DCC configuration that has been created is hidden. The DCC chart created for a block can be permanently deleted via the function **Delete source of the block types...** in the context menu of the library. This command is only available if all blocks in the library have been compiled without errors. If it is executed, all the corresponding chart sources are deleted leaving only what has been compiled in the project. You cannot undo the action. By right-clicking on the DCC chart and selecting the **Block types** command, however, you can open a dialog in which the block libraries used can be exchanged.
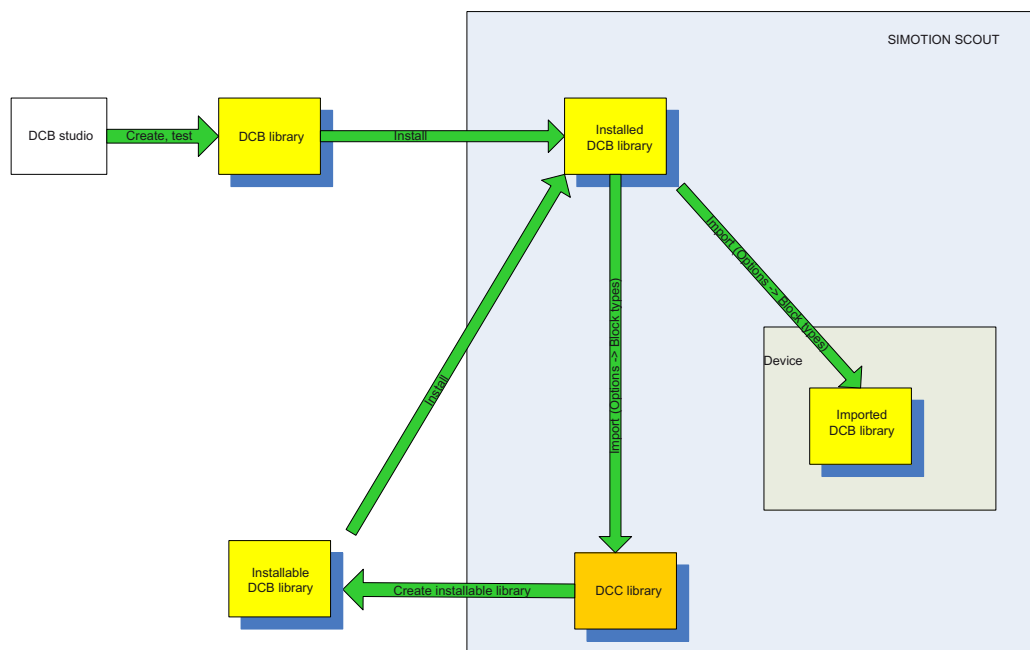
Figure 3-40   Installation of DCB libraries

Requirements

You have already created a new DCC library in the SIMOTION SCOUT engineering system. SIMOTION DCC libraries are located in the folder **Libraries** of the project navigator, DCC libraries in the **SINAMICS Libraries** folder of the project navigator.
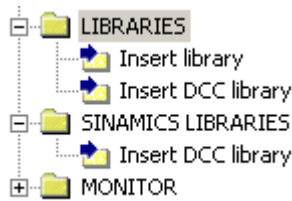
Figure 3-41   Inserting a DCC library

There is already a DCC available for saving in the DCB library.

The chart connections are displayed.

Note

You can display the chart connections via the ⊞ button in the toolbar.

You can use a library either in DCC charts for SINAMICS or for SIMOTION.

Create DCC libraries for SIMOTION in SCOUT in the **Libraries** container.

As of DCC 2.1 / SCOUT / STARTER V4.2, DCC libraries can also be created for SINAMICS.

For this purpose, the project navigator contains a new program container for SINAMICS libraries under **Libraries -> SINAMICS LIBRARIES**.

```
kopfup
   Insert SIMOTION device
   Insert single drive unit
   D455
      X142 inputs/outputs
      EXECUTION SYSTEM
      ADDRESS LIST
      GLOBAL DEVICE VARIABLES
      AXES
      EXTERNAL ENCODERS
      PATH OBJECTS
      CAMS
      TECHNOLOGY
      PROGRAMS
      SINAMICS_Integrated
   LIBRARIES
      Insert library
      Insert DCC library
      DCCLib_1
         DCC_1
         Insert DCC charts
   SINAMICS LIBRARIES
      Insert DCC library
      DCCSinLib_1
         Insert DCC charts
   MONITOR
```

Figure 3-42   Library container for DCC libraries for SINAMICS devices in SIMOTION SCOUT

The DCC libraries for SINAMICS and SIMOTION are distinguished from one another on the basis of the different library containers in the project navigator.

Insert a DCC library for SIMOTION by selecting **Libraries -> Insert DCC library** from the context menu in the project navigator.

Insert a DCC library for SINAMICS by selecting **SINAMICS libraries -> Insert DCC library** from the context menu in the project navigator.

With DCC 2.0, you can insert SIMOTION DCC charts into the library container using copy and paste.

The chart connections of the inserted DCC chart must be connected to the chart connections of the SINAMICS DCC library without BICO interconnections and @ parameters before the compilation.

Delete the DCC chart sources for improved know-how protection.

The DCC libraries, but not the DCB block libraries, are found in the library containers.

By selecting **Generate DCB library...** from the context menu for the selected DCC library, you can create installable DCB libraries from DCC libraries (typical libraries).

It is possible that libraries and block types with the same names are present in the separate directories for the SIMOTION and SINAMICS libraries.

Note

You need a DCC SIMOTION license to create and edit SIMOTION DCC libraries.

You need a DCC SINAMICS license to create and edit SINAMICS DCC libraries.

Rules for creating DCC libraries

- It is only possible to compile charts in DCC libraries.
- The chart may be hierarchical (chart-in-chart).
- Interconnections between charts in the library are not permitted.
- No HMI variables, BICO interconnections, @ variables or @ parameters must be configured at the chart connections of the basic chart and the blocks it contains.
- All blocks may only by integrated into the same execution level / execution group. Here, the chart blocks must directly follow one another.
- Inside the chart folder that needs to be compiled, all of the blocks from a basic chart must be placed in the execution group with the same name as the chart.
- Global operands are not permitted, but must be modeled as an input or output as chart connection.
- For the DCC library and its connections, please select only names starting with lower-case letters.

Procedure

Model your DCC in the DCC editor.

Insert an **ADD**er and a PI controller (**PIC**), for example, in the chart and interconnect them. See Interconnection to chart connections (Page 42)
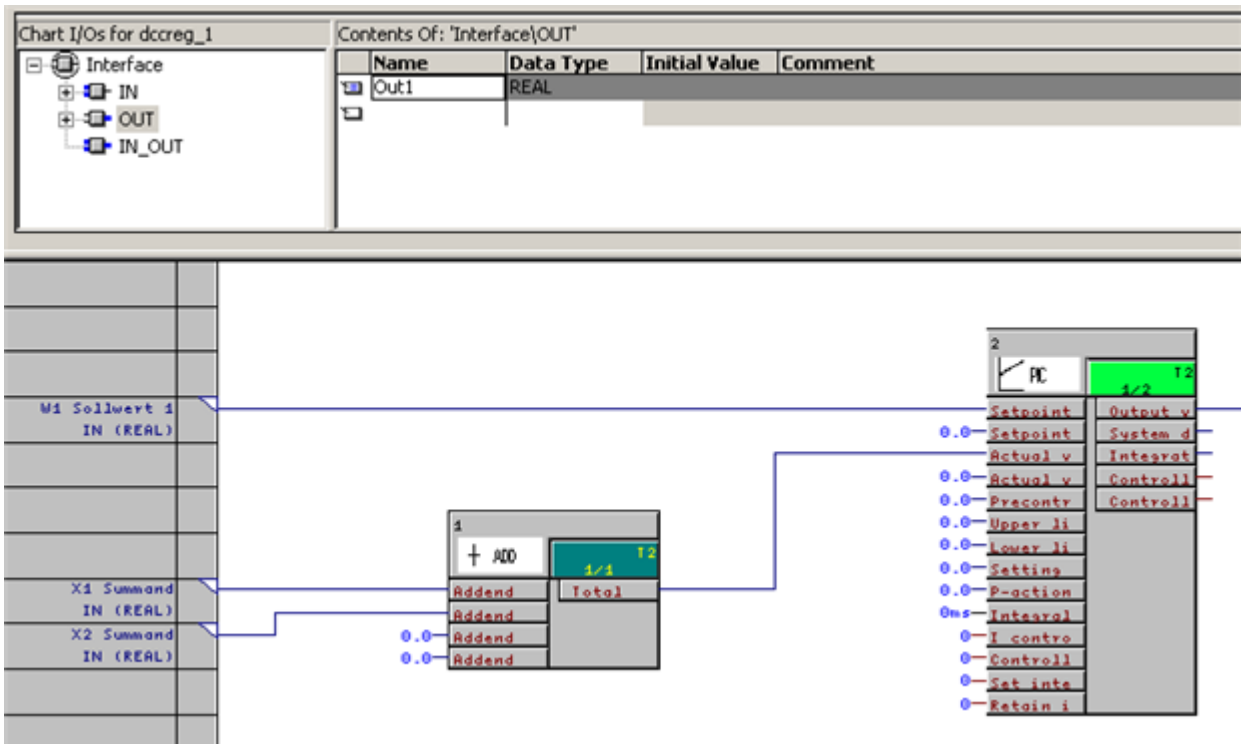
Figure 3-43   Example: Creating a chart as a program

Following successful configuration, the DCC can be compiled as a program. To do this, select **Chart > Compile > Chart as program** in the menu.

The newly created DCB library is available in the menu under **Options > Block types** in the DCC editor. The DCB library is not shown in SIMOTION SCOUT/STARTER; only native libraries are listed there.

See also

Creating an installable DCB library from DCC libraries (Page 96)

3.11.2   Inserting and programming block types in DCC libraries

Within a library, a DCC chart can be created for each DCC block in the library. A new block type is created using the **Insert DCC chart** context menu.

SIMOTION and SINAMICS libraries are both subject to the same rules for assigning DCC chart names:

● Max. 22 characters, beginning with a letter

● If an underscore is used, only numbers may be used after it.

The following rules apply to the blocks:

- There is no restriction on the number of blocks for a DCC library.

- The name of the block type is formed on the basis of the chart name

- The block inputs and outputs are created in the DCC editor using **View -> Chart I/Os** and interconnected with the inputs and outputs of the block instances

- Within a library, all the blocks must be within the same execution group. This group bears the name of the chart

- The execution group in which the blocks run in the target device depends on the execution group of the block instance that calls the library block.

3.11.3 Special features of the 1:N interconnection of chart connection inputs

If you have defined an interconnection of a chart connection input with blocks for a chart, there are two different scenarios. First, there is the scenario of a 1:1 interconnection, which can be carried out without problems. However, the following must be noted for a multiple interconnection (1:N) of a chart connection input to several blocks.
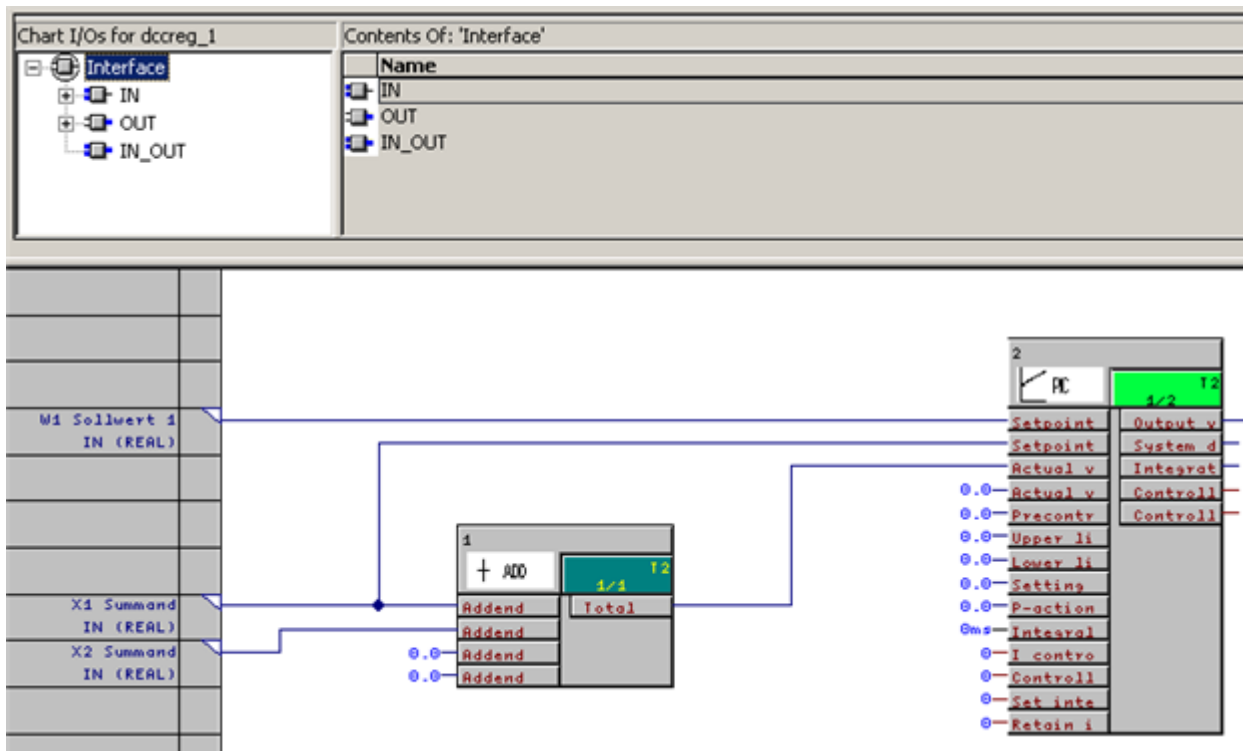
Figure 3-44   Connection example for a 1:N interconnection without NOP block

If there is a 1:N interconnection at an input within a hierarchical chart or block type, an **NOP_x** block must be used in the chart for consistency reasons - refer to the following example:
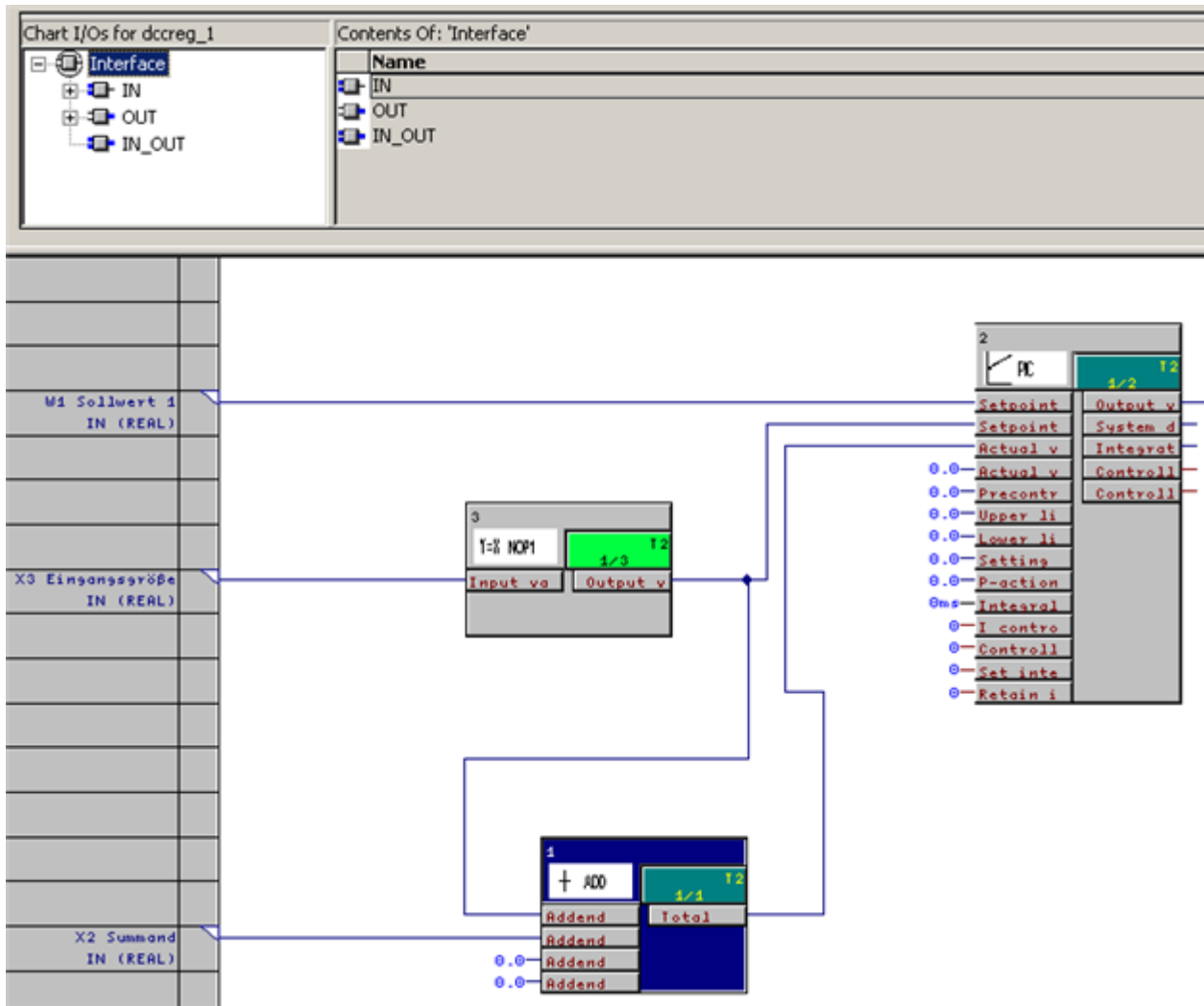
Figure 3-45   Connection example for a 1:N interconnection with NOP block

3.11.4   Creating comments and icons for DCC libraries you have created yourself, and assigning a block family to them

In the engineering system, you can save additional information on individual blocks of a DCB library that you have created yourself.

Procedure

Select the block library you want to edit in the engineering system.

In the context menu of the DCC library, select the **Block type properties** command.
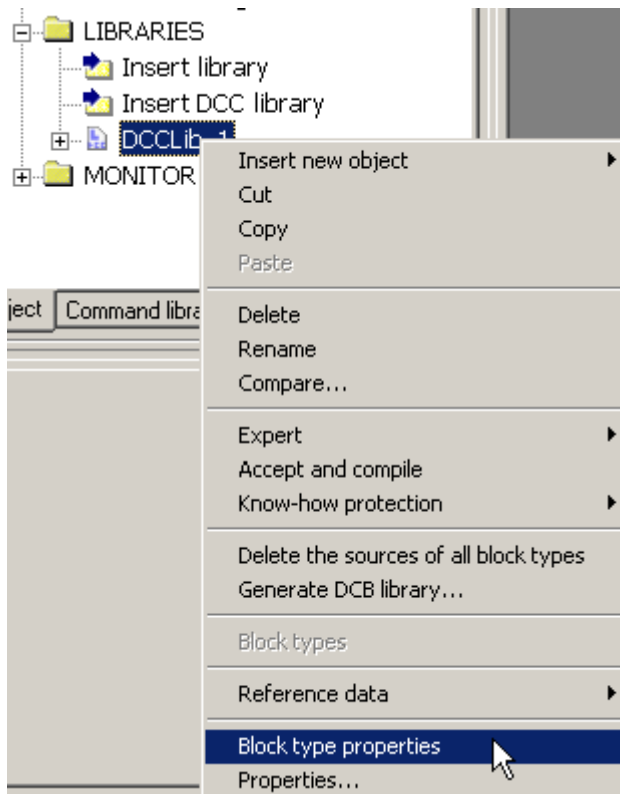
Figure 3-46   The **Block Type Properties** dialog opens.
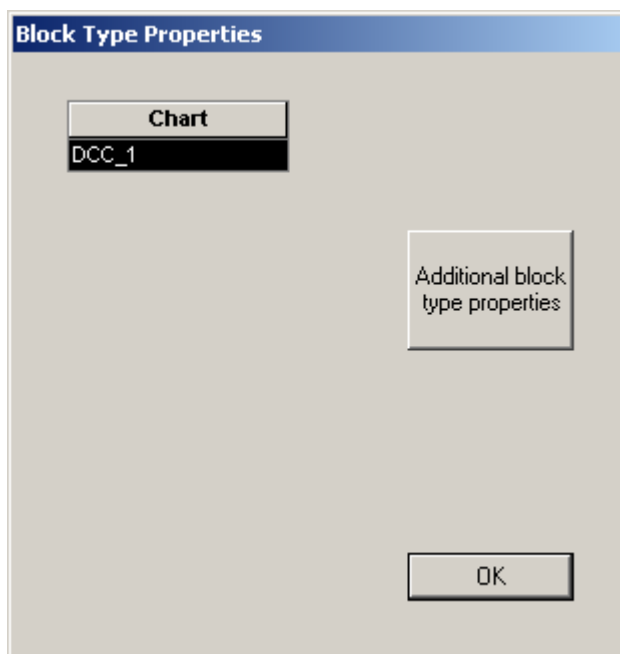
Figure 3-47   Block Type Properties

In the list under **Chart**, select the DCC chart you wish to edit and click the **Additional block type properties** button to open the **Block Type Properties** dialog.
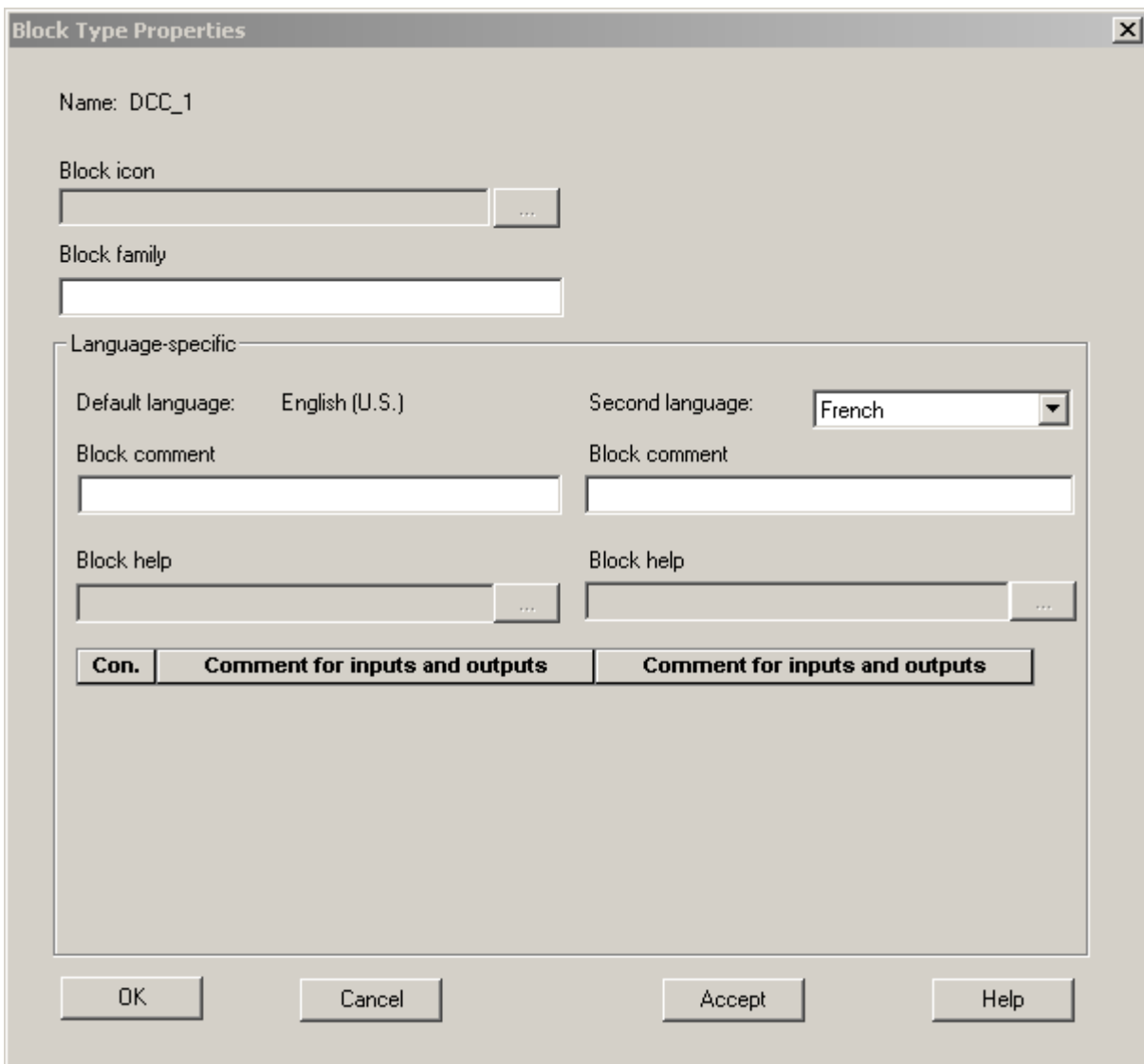
Figure 3-48   Block Type Properties

Block icon

Click the **...** button and load a BMP file you wish to use as a block icon for the DCC chart.

Block family

Enter the family to which you want to assign the DCB library. You can either choose an existing block family (e.g. Logic, System, Conversion, Arithmetic, etc.) or enter a new one, which will then be created automatically. The default language for block families is English.

Block comment

Here you have the option of entering a comment for your DCB library; this will appear in the block header. On the left-hand side, enter the comment in English (the default language). On the right-hand side, you have the option of entering the comment in additional languages. To do this, select the required language from the combo box above and to the right of the field, enter the comment, and click Accept. Repeat this process for all other languages in which you wish to create the comment.

Accept what you have entered using the **Accept** button and click **OK** to close the dialog.

3.11.5 Creating an installable DCB library from DCC libraries

Up to SIMOTION V4.1.1, a DCC library had to be transported as project. As of SIMOTION V4.1.2 / STARTER 4.2, it is possible to create an installable DCB library from a DCC library.

To do this, select the entry **Create DCB library...** in the context menu of the library; the **Create DCB Library** dialog appears.
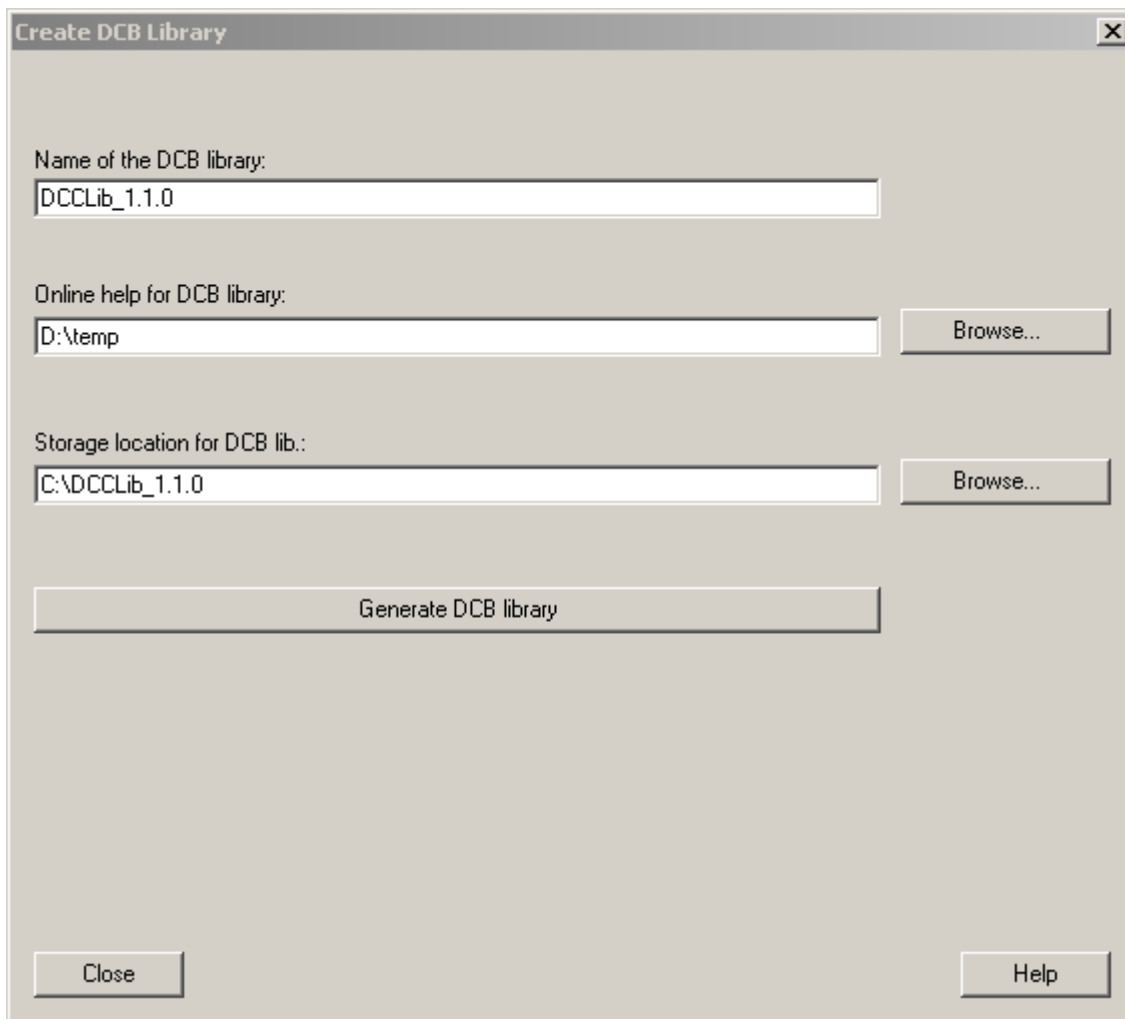
Figure 3-49   Create DCB library

In the dialog box, specify the online help directory for the library.

The name of the DCB library results from the name of the selected DCC library. You can freely select the storage location of the DCB library and then start the creation via the **Create DCB library** button. Information on the creation of the DCB library can be found in the detail view.

You have the option of deleting DCC chart sources for improved know-how protection.

Select the context menu Delete source of all block types in a compiled DCC library to delete the sources of the DCC charts. The charts can then no longer be edited.

By selecting the context menu **Block type -> Properties** at the library, you can allocate comments to block types and block connections in multiple languages.

An installable library may be installed on a different SCOUT/STARTER, regardless of the STEP7 project being used. This library does not contain the CFC chart sources.

Once you have installed the library using **Options -> Install libraries**, it is available for importing into DCC charts on the devices (Options -> Block types).

Note

If you create an installable library from a DCC library, you must replace the DCC library with the installable library in the DCC chart via the menu **Options -> Block types** of the CFC editor.

Note

The sources of the DCCs are no longer contained in the created DCB library. The contained functions can no longer be changed or monitored (know-how protection).

Information on how to create an online help for the DCB library created can be found in the section titled Creating the online help for block libraries (Page 98).

Note

It is possible to create C block libraries from exported DCC libraries automatically. This results in improvements to the memory utilization in the execution performance of the DCC charts as well as increased know-how protection.

See also

Exchanging the basic library version for installed libraries (Page 83)

3.11.6 Creating the online help for block libraries

Note

From SCOUT/STARTER 4.4 the block help editor is no longer part of the setup. A help can now be created as part of SIMOTION CLIB Studio or SINAMICS DCB Studio.

You can create a separate online help for the created libraries with the aid of a supplied editor, the **DCB help editor**. You edit your help files in the same view in which they are later displayed.

The DCB help editor can be found at *...\Siemens\Step7\S7BIN\helpeditor\DCB-Help-Editor.exe*.

Storage structure for online help

Save your online help files under the name **<block>_doc.xml**, whereby <block> is the name of the written block library. Create a folder structure comprising a **help** folder, containing the **help_a**, **help_b**, **help_c**, **help_d** and **help_e** folders. This results in the following assignment:

Table 3-6   Storage structure of the online help files

Folder	Subfolder	Language of the help file <block>_doc.xml
help	help_a	German
	help_b	English
	help_c	French
	help_d	Spanish
	help_e	Italian

A **<block>_doc.xml** file can be stored for each block in the help_x folder. For example, save your German language **<block>_doc.xml** file under **help > help_a**.

Procedure

1. Open the DCB help editor with a double-click on *helpeditor DCB-Help-Editor.exe*.

2. The STARTFILE_DOC.XML file, which is filled with elements and dummies, opens automatically.
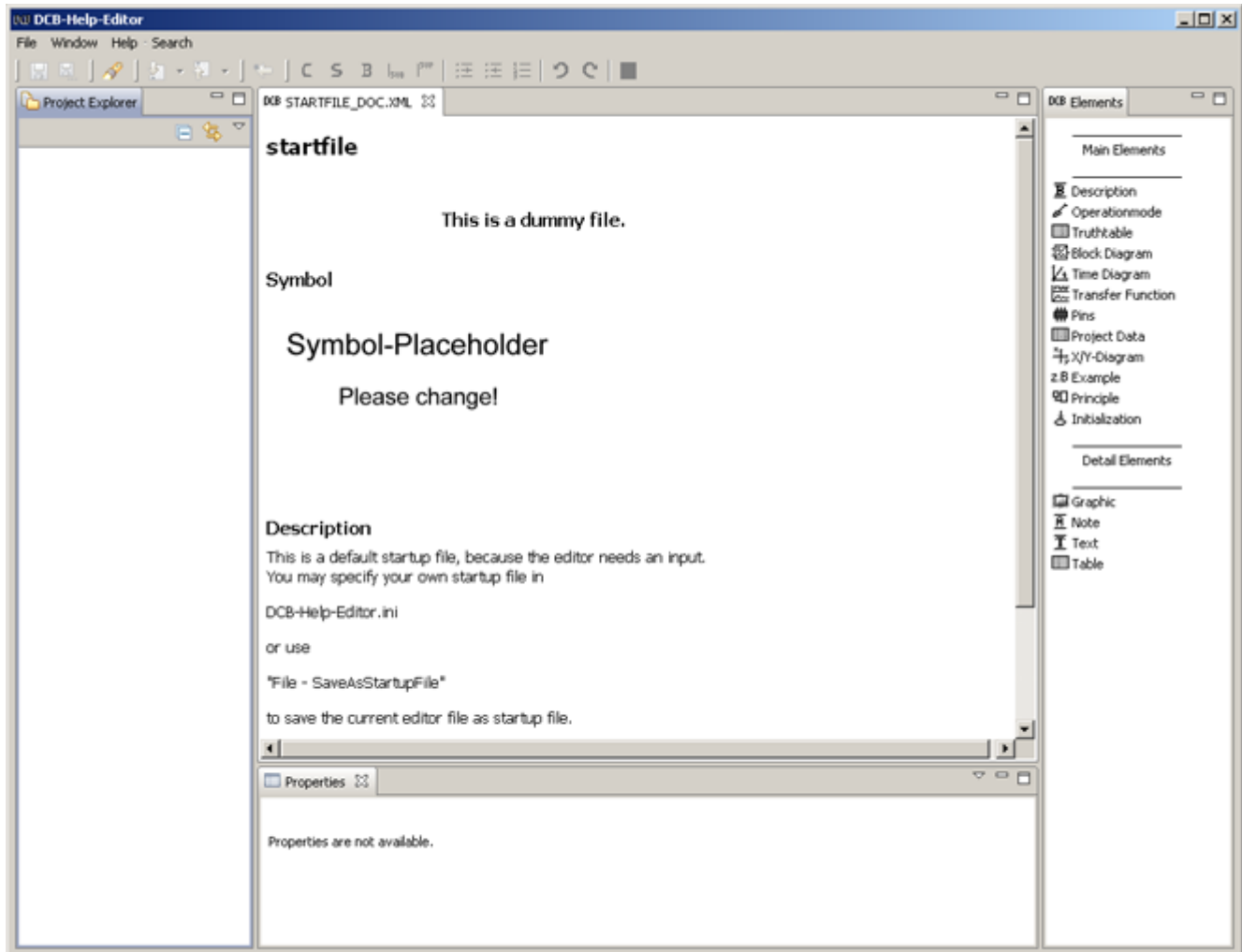
Figure 3-50   DCB help editor with STARTFILE

3. Replace these dummies with your own descriptions by selecting the dummies and entering or copying your text in the file.

4. The **Elements** pane on the right-hand side contains additional elements for the description (e.g. *Truth table*, *Block diagram*, *Example*, etc.), which you can drag to the desired position on your file using drag-and-drop and edit there.
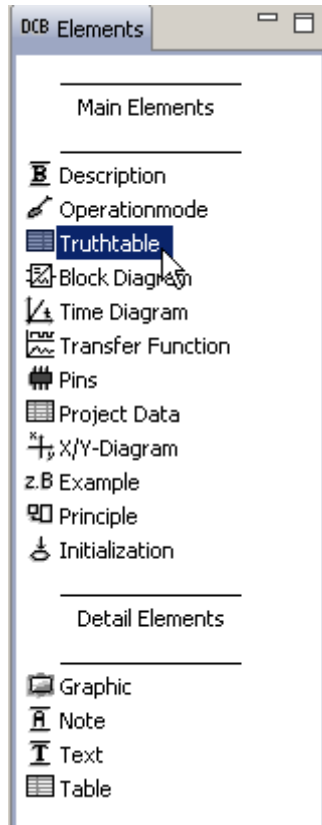
Figure 3-51   Elements for the description

5. Insert your own graphics by dragging the *Graphic* element to a suitable position in the file using drag-and-drop. In the following dialog box, enter the path under which the graphic is stored (use the **Browse** button).
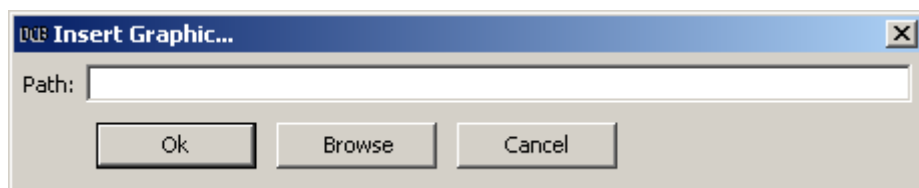
Figure 3-52   Inserting a graphic

6. Save the edited file with the 🖫 button or via the **File > Save as...** menu in the storage structure described above.

Creating further online help files

You can open existing files via the **File > Open** menu.

You can create a new block description via **File > New**.

Note

Detailed operating instructions for the **DCB help editor** can be found in the menu under **Help > Help contents**.

3.11.7   Installation and uninstallation of DCB libraries

As of DCC Version 2.0.2, you have the option of installing or uninstalling DCB libraries from SIMOTION SCOUT using a dialog. As of DCC Version 2.1, you can install SINAMICS DCB libraries and SINAMICS DCC libraries in STARTER/SCOUT during operation and then use them in DCC charts without having to exit and restart STARTER/SCOUT.

You start the dialog box via the menu command **Options -> Installation of libraries and technology packages ...**
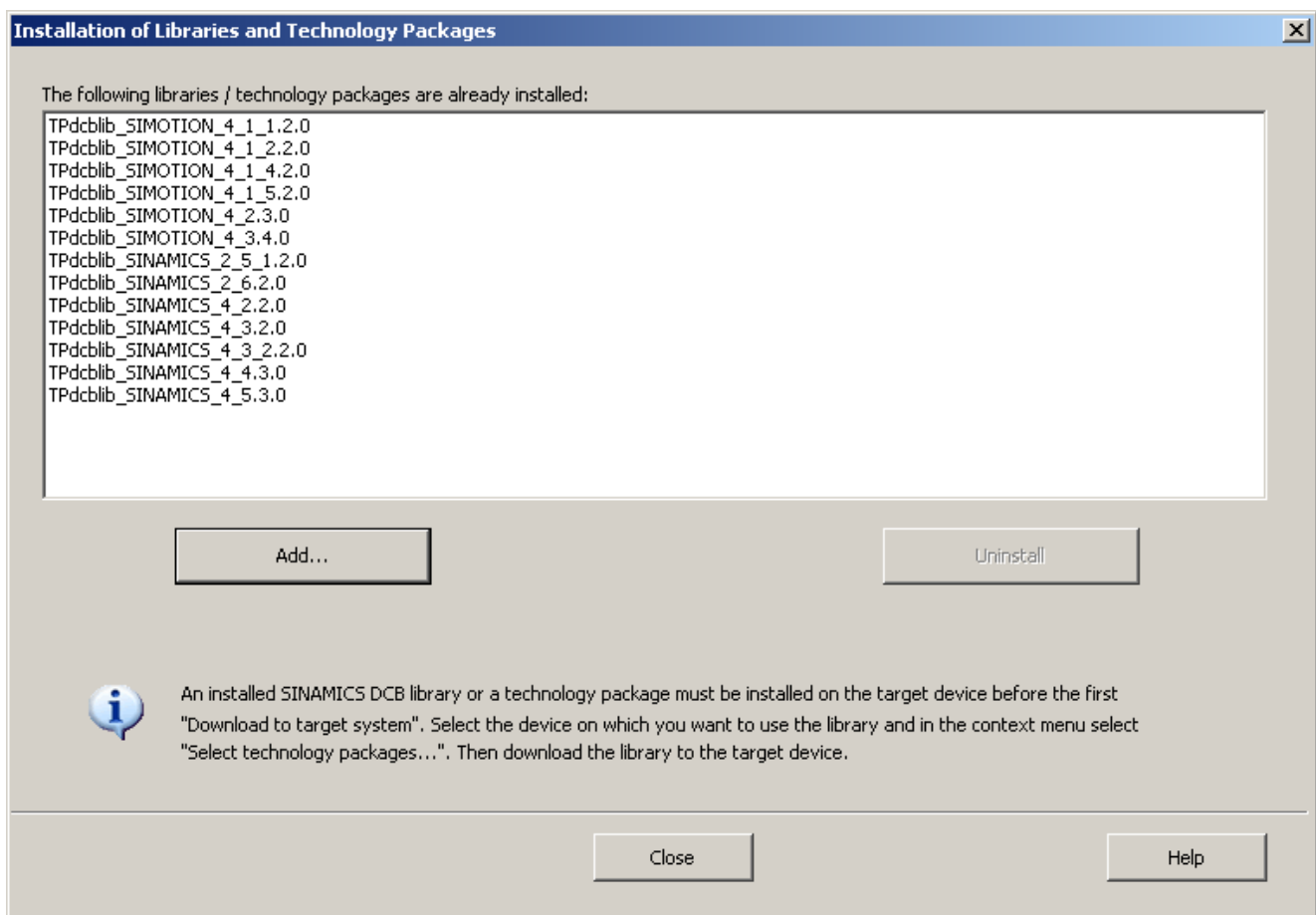
Figure 3-53   Installing and uninstalling DCB libraries - SIMOTION SCOUT

Uninstallation of DCB libraries

Under **Following libraries / technology packages are already installed:** you will see a list of the libraries that have already been installed.

- Select the library to be deleted and select **Uninstall**. The selected library will be uninstalled. A multiple selection is possible. Information on the uninstallation can be found in the detail view.

Installation of DCB libraries

Click the **Add...** button and navigate in the following Open File dialog box to the library to be installed. Select the library and install the selected library by double-clicking or by clicking the **Open** button. A multiple selection is possible. Information on the installation can be found in the detail view.

Exit the **Installation of Libraries and Technology Packages ...** dialog box via the **Close** button.

Note

The libraries are installed independently of the projects in SIMOTION SCOUT/STARTER. The libraries are not transported when projects are archived or exported. If the project is to be loaded on a different SCOUT/STARTER, the libraries used there must be reinstalled. This is also the case after SCOUT/STARTER has been reinstalled.

The DCC standard libraries DCBLIB (supplied from the factory) are preinstalled. If you explicitly require this as an installable library, you can find this library, e.g. "dcblibV2_0_simotion4_1_5.zip" (version-dependent name) after the installation of SCOUT in the"C:\Program Files\Siemens\Step7\U7umc\data\dcc\SIMOTION" directory or, if available, on the DCC DVD under "VOL1\CD_1\DCC\DCC_DCBLIB_SIMOTION\Disk1". You can also, for example, store the standard DCC libraries using the card reader in the USER directory on the CompactFlash card for later service assignments (the version of the standard DCC library may not be available in the engineering tool). This means that if service is required, you can use the CompactFlash card reader and the functions described here to reinstall the relevant libraries.

Note

For SINAMICS libraries as compared with SIMOTION libraries, there is the restriction that DCBLIB standard libraries can only be installed and uninstalled while no project is open.

The libraries are installed independently of the projects in STARTER. The libraries are not transported when projects are archived or exported. If the project is to be loaded on a different STARTER, the DCC libraries used there must be reinstalled. This is also the case after STARTER has been reinstalled.

The DCC standard libraries DCBLIB (supplied from the factory) are preinstalled. If you explicitly require this as an installable library, you can find this library, e.g. "dcblibV2_0_sinamics2_6.zip" (version-dependent name) after the installation of STARTER in the directory "C:\Program Files\Siemens\Step7\U7umc\data\dcc\SINAMICS" or, if available, on the DCC DVD under "VOL1\CD_1\DCC\DCC_DCBLIB_SINAMICS\Disk1". You can also, for example, store the standard DCC libraries using the CompactFlash card reader in the USER directory on the CompactFlash card for later service assignments (the version of the standard DCC library may not be available in the engineering tool). This means that if service is required, you can use the CompactFlash card reader and the functions described here to reinstall the relevant libraries.

See also

Updating the block library (Page 80)

3.11.8 Exchanging the target device family for DCC libraries

As of SIMOTION SCOUT V4.3 you can exchange DCC libraries between the SINAMICS and SIMOTION target devices. In this way you can use a DCC library available for DCC SIMOTION on a SINAMICS target device or a DCC library available for DCC SINAMICS on a SIMOTION target device.

Note

This function is not available for STARTER.

Proceed as follows to insert DCC libraries in a different target device family:

In the project navigator, select the DCC library to be copied at the appropriate library container and select **Save as DCC SINAMICS library** or **Save as DCC SIMOTION library** in the context menu.

Insert new object ▶
Cut
Copy
Paste
Delete
Rename
Insert as SIMOTION chart
Save as DCC SINAMICS library ▶
Expert ▶
Accept and compile
Know-how protection programs ▶
Delete the sources of all block types
Generate DCB library...
Block types
Reference data ▶
Block Type Properties
Properties...
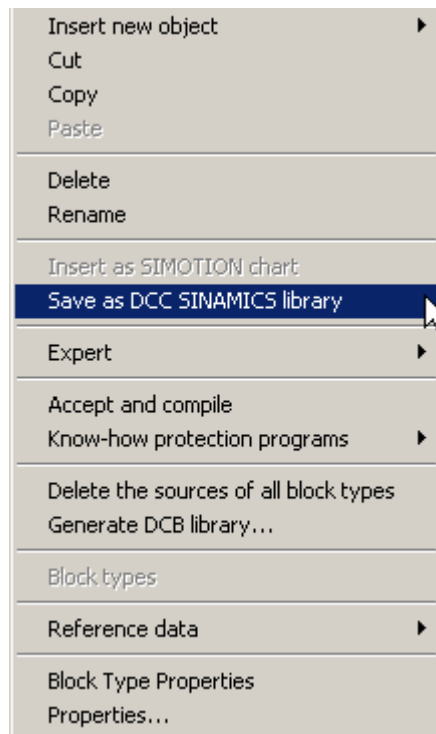
Figure 3-54   Save as DCC SINAMICS library

Proceed as follows to insert DCC library blocks in a DCC library of a different target device family:

In the project navigator, select the DCC chart to be copied on the DCC library at the library container and select **Copy** in the context menu. Switch to the library container for DCC library of the other target device family and insert the DCC chart via the context menu command **Insert as SINAMICS chart** or **Insert as SIMOTION chart**.
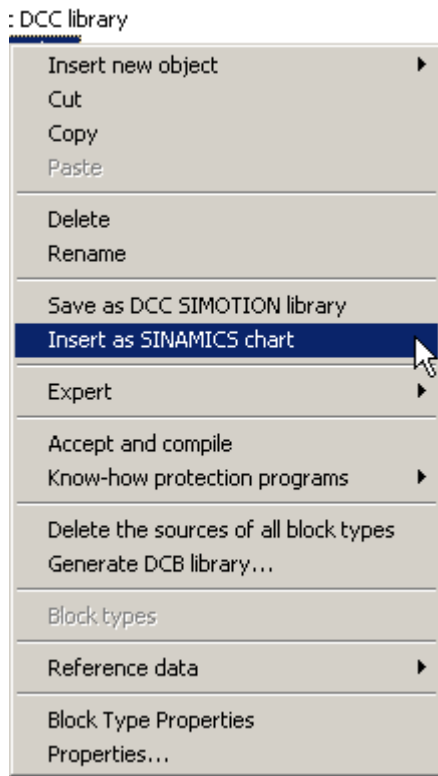
Figure 3-55   Insert as SINAMICS chart

When you insert the chart, the original library/chart name is taken over and an underscore and a number appended to the name. If there is already an underscore and a number appended to the name, the number is incremented.

The execution groups are changed automatically when saving under a different target device family.

Note

If chart sources of the DCC libraries are not available, no DCC editor is installed or there is no license for the DCC editor, you cannot insert DCC libraries into a different target device family. The **Save as** context menu command is then not available.

If block types from the library are not available on the target device, an error is output during compilation of the library. In this case, you must either delete or modify the block instances.

3.12 Know-how Protection

3.12.1 Know-how protection for libraries and programs

The SIMOTION SCOUT/STARTER engineering system provides know-how protection functions for your data. This enables you to prevent unauthorized access to your programs/ parameters in the drive unit. There are two different types of know-how protection:

● Know-how protection for programs (ST source files, MCC charts, LAD/FBD programs and DCC charts) and libraries

● Know-how protection for drive units (as of SINAMICS V4.5)

Know-how protection for programs

The know-how protection installed in SIMOTION SCOUT protects the programs (ST source files, MCC charts, LAD/FBD programs and DCC charts) and libraries in your project.

In the project navigator, the protected programs contained in the project are displayed grayed-out. You can only open these programs for editing after you have entered the password. The know-how protection is retained after copy/paste and also after export/import.

Know-how protection for programs - setting-up/deleting a standard login

If you have not yet set up a standard login for the know-how protection, open the **Edit Standard Login** dialog box via the menu **Project -> Know-how protection for programs -> Edit standard login...** and enter a standard login there. Assign a password, confirm this and close the dialog box with the **OK** button.
You can delete the standard login in this dialog box. A password is not required for this.

Configuring know-how protection for programs

Call the **Configure Know-how Protection for Programs...** dialog box via the menu **Project -> Know-how protection for programs** and select the desired coding type for programs and libraries.

Note

The dialog box is only available in SIMOTION SCOUT.

Standard (libraries and programs)

This level corresponds to the old security standard. Access to the programs is protected by a login and password. Programs and libraries can be recompiled at any time without knowing the password. This level is completely downward compatible, i.e. you can edit the project in

an older version of SIMOTION SCOUT by saving the project in the old project format or by exporting and importing the project again.

Note

Only the standard coding can be used for DCC sources and libraries.

Medium (libraries and programs)

The coding of the password has been improved. Programs and libraries can be recompiled at any time without knowing the password. However, because of the changed coding, downward compatibility is no longer possible without knowing the password. If, for example, the program is exported and imported into an older SIMOTION SCOUT version or uploaded to a SIMOTION SCOUT project of an older version, the source text can no longer be displayed there.

High (for ST source files only)

In this level, (re)compilation is only possible after input of the password. However, libraries protected with this level can of course also be used after an export without knowledge of the password, because in this case the compilation result is also exported. Exports and imports are only possible in the same or higher SIMOTION SCOUT versions.

Note

The various levels of the know-how protection only apply to the new programs to be protected. You may have to delete, reconfigure and activate the know-how protection for already protected programs.

Activating know-how protection for programs

1. Open the project and select the PROGRAMS folder or the individual program in the project navigator.

2. Select the entry **Know-how protection for programs > Set** in the context menu.

Deleting know-how protection for programs

1. Select the PROGRAMS folder or the individual program in the project navigator.

2. Select the entry **Know-how protection for programs > Delete** in the context menu.

3. Enter the password in the following **Know-how Manager** dialog box and close the dialog with the **OK** button.

See also

Know-how protection for drive unit (Page 109)

3.12.2   Know-how protection for drive unit

As of SINAMICS V4.5, know-how protection is available for drive units in STARTER / SIMOTION SCOUT. You can only activate/deactivate the know-how protection in online mode. When the know-how protection is set, p parameters can be neither read nor written. If certain parameters are to be excluded from the know-how protection, enter them in the expert list under parameter p7764 before you set the know-how protection.

Know-how protection for drive units - default setting

The know-how protection may already have been activated for drive units as soon as you have inserted a drive unit. In this case, consult the documentation for the drive unit.

If you want to change the pre-assigned password for the know-how protection, you must first deactivate the know-how protection via the context menu **Deactivate know-how protection**.

To do this, select the option **Definitive (password will be deleted)** in the **Deactivate know-how protection for drive unit** dialog box.

Enter the default password and close the dialog box with the **OK** button.

Activating know-how protection for a drive unit

When you activate know-how protection for the drive unit, a dialog for entering a password always appears. You can activate the SINAMICS know-how protection for drive units either with or without copy protection.

The **Know-how protection without copy protection** option is active by default.

You can select the following copy protection options for devices as of firmware V4.7:

- Basic copy protection (linked to the memory card)

- Extended copy protection (linked to the memory card and the Control Unit)

Only the **With copy protection** option is available for devices up to firmware V4.6; this corresponds to the extended copy protection.

If no know-how protection for the drive unit has been activated, but a password has already been defined (temporarily deactivated know-how protection for the drive unit), you can reactivate know-how protection for the drive unit with the previous password. You can, however, also change the password.

Define password

To define the password, you need to enter and confirm the new password. If you make a mistake, a message appears to inform you that the passwords entered do not match. If this happens, you need to reenter the passwords. Passwords are subject to the following rules:

- Min. length: 1 character, max. length: 30 characters.

- All writeable characters are permitted, see Opening a project with any language setting under Windows (language-neutral).

- It is the personal responsibility of the user to ensure adequate password security.

Opening a project with any language setting under Windows (language-neutral)

If you want to edit your projects in different Windows systems with different language settings, you should only use characters from the ASCII character set to ensure problem-free working with know-how protection. As a general rule, a project must first be enabled for this purpose in the SIMATIC Manager, but independently of the know-how protection. To do this, open the project in the SIMATIC Manager. Then go to the Edit menu and select the following option under Object properties: Can be opened with any Windows language setting (language-neutral).

Note

If you want to process your projects on Windows systems with different language settings, then you must define this in the project settings. Navigate via the menu **Project -> Properties** to the **Properties - Project** dialog box and activate the **Can be opened under every Windows language setting (language-neutral)** checkbox. Only use ASCII characters for the password of the know-how protection for drive units.

Diagnostics functions

If, despite active know-how protection, you still require the trace, measuring function or the function generator, select the **Permit diagnostic functions** option. You may have to activate the required parameters first in the exception list.

Note

Diagnostic functions possible as of V4.7

Diagnostic functions that work despite active know-how protection are available as of V4.7.

Deactivating know-how protection for a drive unit

To deactivate the feature, you must enter the correct password. The know-how protection for the drive unit can be:

- **temporarily** deactivated: The know-how protection is active again after switching off and switching on.

- **permanently** deactivated: The know-how protection remains deactivated even after switching off and switching on again.

If you select "Permanently", you can also carry out a data backup on the Control Unit with **Copy RAM to ROM**. The checkbox with the same name is active in this case and is automatically activated. If you deactivate this checkbox, you must perform a manual **RAM to ROM** data backup later if the know-how protection remains deactivated after switching off and on.

Absolute know-how protection for the drive unit

Absolute know-how protection that cannot be deactivated is created by deleting parameter p7766 from the exception list and then activating the know-how protection. Create a backup copy of your project before you activate absolute know-how protection.

Combining know-how protection for drive unit with write protection for drive units

You can combine the **Know-how protection for drive unit** with **Write protection for drive units**. Note that you must first set the **Know-how protection for drive unit** and then the **Write protection for drive units**.

If you have combined the **Know-how protection for drive unit** with the **Write protection for drive units**, the parameters of the exception list can only be read. The **Write protection for drive unit** does not apply for parameters with the WRITE_NO_LOCK attribute. These parameters can be written at any time. With combined write and know-how protection, the drive unit behaves similar to absolute know-how protection for drive unit, since a password can no longer be entered due to the write protection.

Functions available for activated know-how protection for drive units

When know-how protection for the drive unit is activated, the following functions remain available:

- Go online
- Restore factory settings
- Get information about the firmware version
- Read out errors (drive status, warnings, faults), alarms, the diagnostics buffer, and the alarm history
- Acknowledge errors/alarms
- Control panel functions
- Display created acceptance documentation
- Upload (restricted scope)

When know-how protection for the drive unit is activated, the following functions are not available:

- Download
- Export/import
- Trace
- Function generator
- Measuring functions
- Automatic controller setting
- Stationary/rotating measurement
- Delete alarm history
- Create acceptance documentation

The following functions are optional:

- Trace
- Function generator
- Measuring functions

These functions can only be executed if the diagnostic functions are permitted when the know-how protection is activated.

Behavior for devices with know-how protection for the drive unit

- Know-how protection always applies to the device, not to individual DOs.

- The content of the project navigator remains unchanged.

- All r parameters can still be read.

- p parameters can be neither read nor written with the following exceptions:

  - Parameters with the KHP_ACTIVE_READ or WRITE_NO_LOCK attribute

  - Parameters from the exception list

- Instead of a complete expert list, only a reduced expert list is displayed. This contains all the r parameters as well as the p parameters that have been enabled by attributes or the exception list. The behavior also applies to functions of the expert list, user-defined lists of values, and watch tables.

- An expert list comparison can be performed, the values of the know-how-protected parameters are displayed as **know-how protected**.

- No wizards are available.

- It is not permissible to delete this device in its entirety or to delete individual components from it.

- It is not permissible to rename the device.

- It is not possible to call dialogs containing functions that are not supported by know-how protection for the drive unit. An error message is displayed.

- Where windows contain functions that are not supported by the know-how protection for the drive unit, they do not show any active contents (with the exception of the expert list and version overview); instead, they indicate that know-how protection for the drive unit is activated.

- Scripts which need read or write access to protected parameters/functions, cannot grant appropriate access and are rejected.

See also

Know-how protection for libraries and programs (Page 107)

Write protection for drive unit (Page 117)

3.12.3 Setting know-how protection for a DCC chart

Setting the know-how protection function for a DCC chart

A DCC chart can be protected as follows:

1. Select the chart in the SIMOTION SCOUT/STARTER.

2. In the context menu of the DCC chart, execute the **Know-how protection -> Set** function in offline mode.
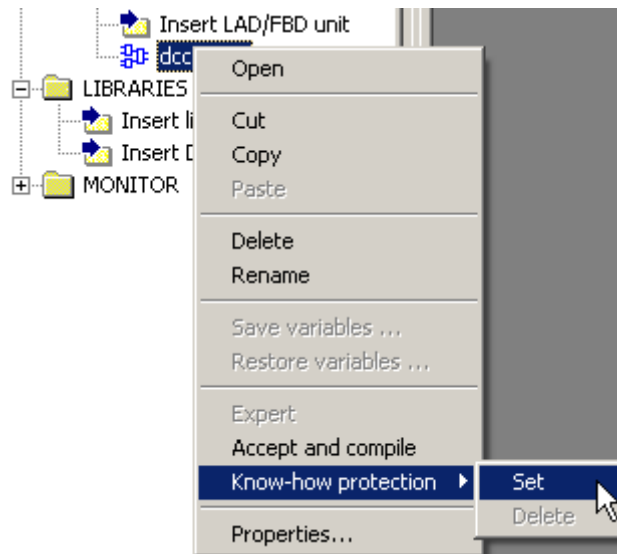
Figure 3-56   Setting know-how protection for a DCC chart

3. When using the know-how protection function for the first time, you will be prompted to enter the required access data (login & password) for protection of the chart in the **Know-How Manager** window. Close the **Know-How Manager** window via the **OK** button.
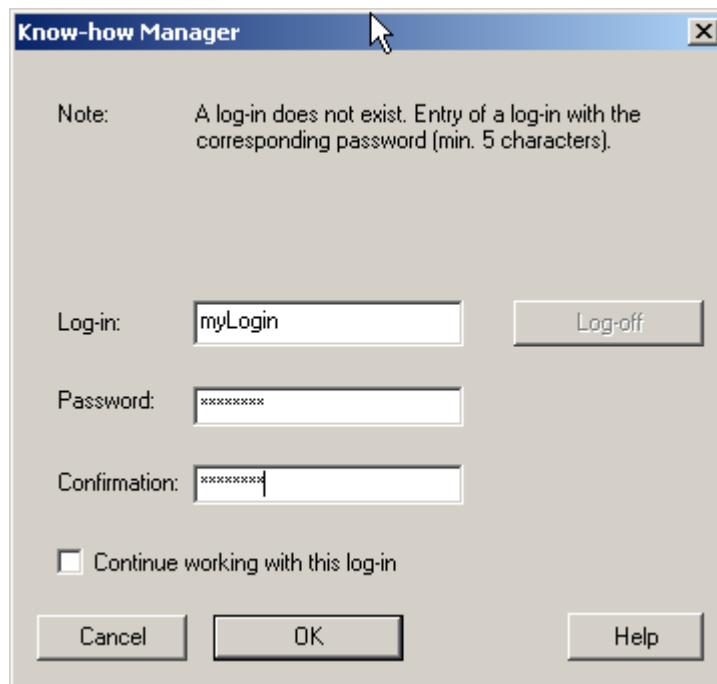
Figure 3-57   Know-how manager

The DCC chart is now shown grayed out.

You can also define a general login with password in the **Project -> Know-how protection** menu item, which then is used automatically for setting the know-how protection on the selected chart. Although you are then automatically logged in with this login, you can at anytime log off from the **Project -> Know-how protection** menu item.

Activating the know-how protection function

To activate the know-how protection, you must log off the logged-in user in the **Know-How Manager** window (access to the **Know-How-Manager** is via **Project -> Know-how protection** in

the menu). The protected DCCs of the opened project are then locked and cannot be opened. Only by entering the password can you remove the lock and open the DCCs.
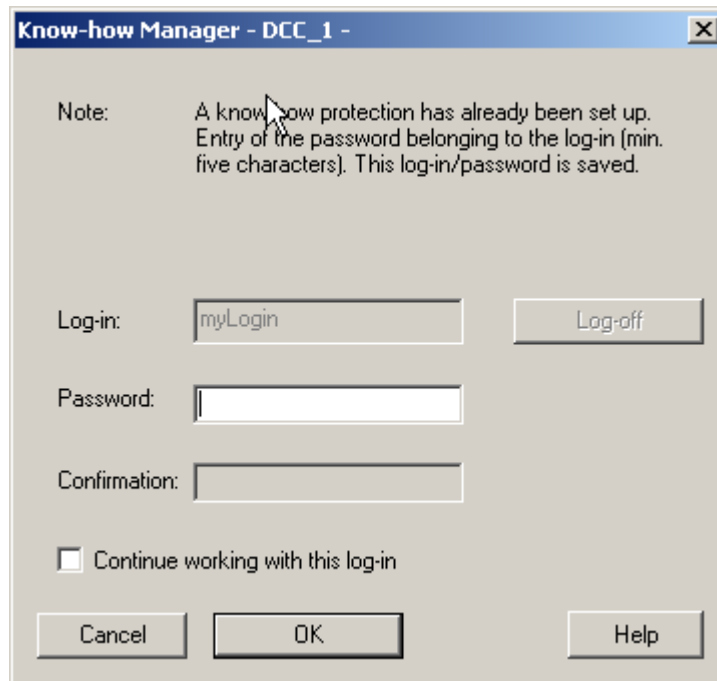
Figure 3-58　　Know-how manager

Note

When DCC charts are protected, the execution sequence cannot be protected in the execution system. Although the chart can be moved, it cannot be edited. Note that if the execution sequence of a protected DCC chart is changed, the functionality can no longer be guaranteed or may fail.

Deleting the know-how protection function

To delete the know-how protection, you must first be logged in (if you are not already logged in, you are automatically prompted to do this during the procedure described below).

You can delete the know-how protection of a chart as follows:

1. Select the chart in the SIMOTION SCOUT/STARTER.

2. In the context menu of the DCC chart, execute **Know-how protection -> Delete**.

3. Close the **Know-How Manager** window via the **OK** button after entering the password.
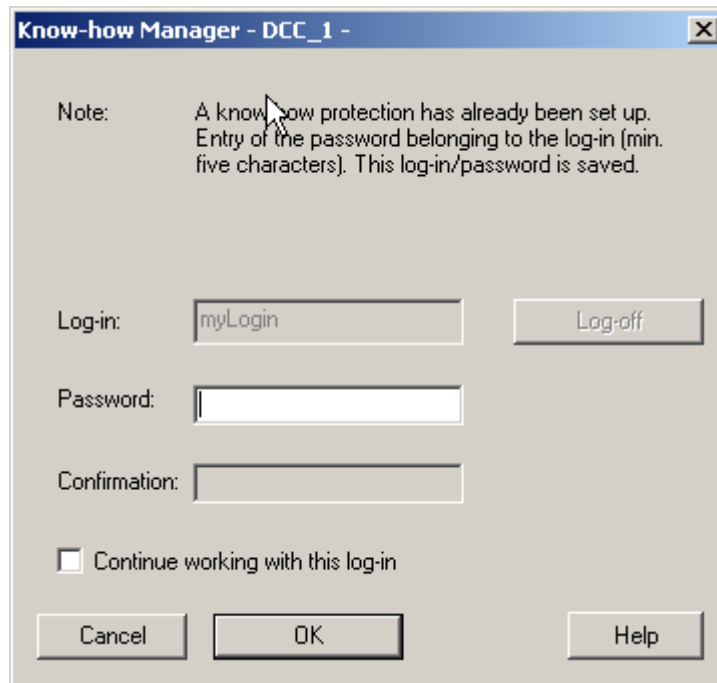
Figure 3-59   Know-how manager

Note

The know-how protection is not automatically reactivated in SIMOTION SCOUT/ STARTER when you close the DCC chart.

Canceling the know-how protection

If you want to open a know-how-protected DCC chart, you can do this by double-clicking as usual. If you are already logged in, the know-how protection is automatically canceled, if you are not logged in, you are automatically prompted to do so when opening.

3.13   Write protection for drive unit

As of SINAMICS V4.5, write protection for drive units is available. You can only activate/ deactivate this write protection in online mode.

Activate/deactivate the write protection via the context menu **Write protection for drive unit** of the drive unit or via parameter *p7761* of the expert list. A password is not required to activate/ deactivate the write protection. You can query the status of the write protection via parameter*r7760.0* in the expert list (value = 0 means that the write protection is not set).

When write protection is activated, the following functions can be used in offline mode:

● Inserting, editing, deleting DCC libraries

● Inserting, editing, deleting DCC charts

● Inserting, editing, deleting DCC blocks

Note

When write protection is activated for a SINAMICS target device, however, a download to the target device is not possible. This prevents any changes made offline in DCC charts taking effect in the target device.

The following functions cannot be used when write protection is activated:

● Editing a DCC chart in test mode

● Inserting or deleting blocks online

● Inserting, moving or deleting connections

● Compiling a DCC chart

● Saving and recompiling all

● Downloading the project

With active write protection, compilation of the DCC chart on the DO is rejected both from the workbench as well as from the DCC editor with the following error message:
"Write protection is currently active on the drive unit. The DCC chart cannot be compiled."

Write access to DCC parameters is rejected with the following error message:
"Value was rejected.
Parameter xxx could not be set to this value. The old value will take effect again."

Note

If you want to activate the write protection permanently, execute the **Copy RAM to ROM** function.

See also

Know-how protection for drive unit (Page 109)

Know-how protection for libraries and programs (Page 107)

3.14   Startup behavior

With regard to the startup behavior, note that the respective initialization value is active at the block connections during startup and then the calculated value in each following cycle.

For more detailed information on the start-up behavior, please see Chapter Characteristics of block connections (Page 38).

3.15   Software upgrade and module exchange

SCOUT or STARTER projects that contain DCC charts and are compiled with V4.1.1 or V4.1.2 can be opened and loaded with SCOUT/STARTER V4.1.3, even without a DCC license. This applies regardless of whether or not the DCC chart sources (i.e. original project) are available.

Note

This specification applies as long as the device version does not change. Following a replacement with a new device version, the DCB library versions present on the new device must be imported and the DCC chart recompiled. The corresponding DCC license (CFC for SIMOTION/SINAMICS) is required for this.

Until DCC 2.1, the library version had to be updated manually after the upgrade of a device version.

As of DCC 2.1 / SCOUT/STARTER V4.2, the library version is upgraded automatically.

Saving and restoring retain variables - SIMOTION only

DCB block instances may contain retain variables.

Select **Save variables...** from the context menu of the DCC chart to save the contents of these variables to a file. The values can then be restored from this file. DCC retain data is retained when the platform is changed or the version is upgraded.

3.16 Version information

Proceed as follows to display the version information:

1. Left-click the *Help -> Info...* menu item in STARTER/SCOUT

2. Click the *System information...* button in the following *Information* dialog
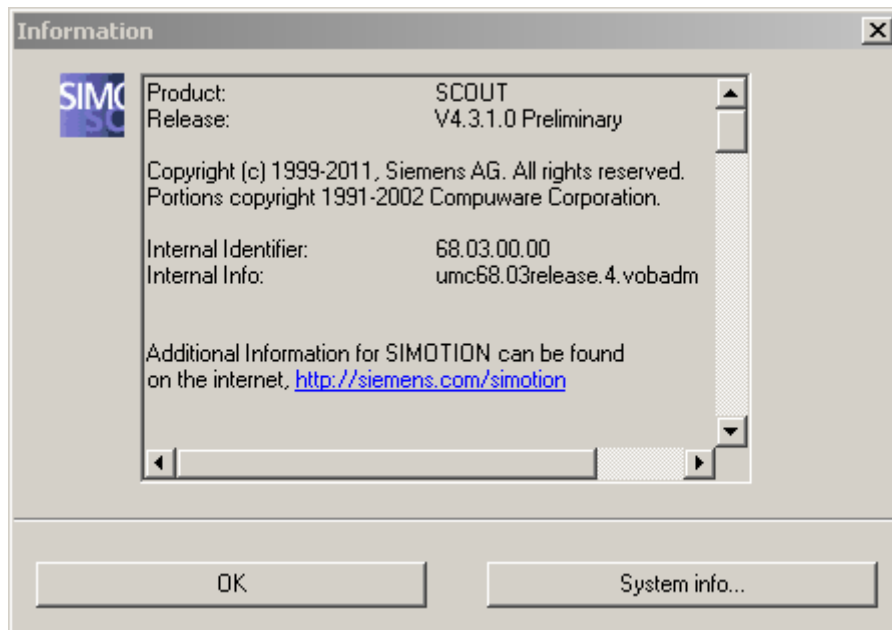
Figure 3-60   Version information - Information

3. The *System Information* dialog with the system information appears.

System Information

Product	Version	Release	Compatibility	Comment
Automation License Manager	5.1.1.0	K05.01.01.00_01.07.00.05	OK	
dcblibV2_0_simotion4_1_1	2.0.25.0	V02.02.01.00_68.03.00.00	OK	
dcblibV2_0_simotion4_1_2	2.0.115.0	V02.02.01.00_68.03.00.00	OK	
dcblibV2_0_simotion4_1_4	2.0.113.0	V02.02.01.00_68.03.00.00	OK	
dcblibV2_0_simotion4_1_5	2.0.43.0	V02.02.01.00_68.03.00.00	OK	
dcblibV2_0_sinamics2_5_1	02.50.32.09	V02.02.01.00_68.03.00.00	OK	
dcblibV2_0_sinamics2_6	02.60.55.00	V02.02.01.00_68.03.00.00	OK	
dcblibV2_0_sinamics4_2	04.20.23.00	V02.02.01.00_68.03.00.00	OK	
dcblibV2_0_sinamics4_3	04.30.21.02	V02.02.01.00_68.03.00.00	OK	
dcblibV2_0_sinamics4_3_2	04.30.22.00	V02.02.01.00_68.03.00.00	OK	
dcblibV3_0_simotion4_2	03.00.100.00	V02.02.01.00_68.03.00.00	OK	
dcblibV3_0_sinamics4_4	04.40.09.00	V02.02.01.00_68.03.00.00	OK	
dcblibV3_0_sinamics4_5	04.50.03.00	V02.02.01.00_68.03.00.00	OK	
dcblibV4_0_simotion4_3	04.00.111.00	V02.02.01.00_68.03.00.00	OK	
DCC Editor SIMOTION	2.2.1.0	V02.02.01.00_68.03.00.00	OK	
DCC Editor SINAMICS	2.2.1.0	V02.02.01.00_68.03.00.00	OK	
DCC Help System	2.2.1.0	V02.02.01.00_68.03.00.00	OK	
Drive ES BASIC	not installed			not required
Drive ES SIMATIC	not installed			not required
Drive ES SlaveOM	5.8.1.0	V05.08.01.00_68.03.00.00	OK	
SIMATIC CFC	8.0.0.0	V08.00.00.00_01.22.00.02	OK	
SIMATIC Device Drivers	8.2	K08.02.02.00_01.10.00.02	OK	
SIMATIC L7-SiS	4.2.1.0	K4.2.1.0_1.21.0.1	OK	
SIMATIC NET PC Software	not installed			not required

OK

Figure 3-61   Version information - System information

3.17 XML export/import of DCC charts

During the project export, the CFC charts with the STEP7 data storage are exported in binary format.

From DCC Version 2.1, DCC charts can be individually exported in XML format and imported again during XML import.

This means that you can transfer a DCC chart to another project (from a project with SINAMICS 2.6 devices to a project with SINAMICS 4.x devices, for example).

The following restrictions apply to the XML export/import of DCC charts:

- DCC charts that have been exported from SIMOTION devices cannot be imported into SINAMICS devices, and vice versa

- DCC blocks that have been exported from SIMOTION DCC libraries cannot be imported into SINAMICS DCC libraries, and vice versa.

The table below outlines the compatibility of XML exports/imports in relation to the relevant SCOUT/STARTER version.

Compatibility	
Project	Projects exported with SCOUT/STARTER V4.1 can be imported with V4.2 without any loss of information. After opening the project, the user is asked to convert the CFC charts to CFC 7.1.
	Projects exported with SCOUT/STARTER V4.2 can be imported with V4.1 without any loss of information. However, it is not possible to edit DCC charts with CFC 7.0.
	In the case of projects that were created with V4.2 and imported using an older version of SCOUT/STARTER, a warning appears when the import process is carried out.
Device	Devices exported with SCOUT/STARTER V4.1 can be imported with V4.2 without any loss of information. If block types that were not available in V4.1 are used in a V4.2 project, the devices can be imported without any loss of information. Errors are reported when compilation is carried out, however.
	Devices exported with SCOUT/STARTER V4.2 can be imported with V4.1. The DCC chart sources are lost when they are imported with V4.1. No errors or warnings are output.
Drive object (DO)	Drive objects exported with SCOUT/STARTER V4.1 can be imported with V4.2. These do not contain any DCC charts.
	Drive objects exported with V4.2 can be imported with V4.1. The DCC chart sources are not imported.
	If a DCC chart is exported on a DO type A and imported on a DO type B, the adaptation process for the sheet bar interconnection is the same as copying/pasting DCC charts. When pasting on the new DO, the old execution group is also transferred. During compilation, the execution group that applies to the new DO is entered automatically.

Compatibility	
CFC version	DCC charts that were created with an older CFC version can be imported with a newer CFC version. Downward compatibility is also supported.
	The export/import of projects and chart containers only supports upward compatibility. A chart created using CFC 7.1 can no longer be opened with CFC 7.0.

The following requirement applies to the XML export/import of DCC charts:

- The CFC editor must be installed

XML export/import of DCC charts is possible at the following levels:

- Project
- Device
- DO (drive object, CUxx, TBxx, TMxx)
- DCC chart
- DCC library

The assigned DCC charts are included in the XML export/import of the selected object.

Note

If a CFC editor is not installed, only the DCC charts (i.e. without chart sources) are exported/ imported. No error message is output.

No DCC license is required for exporting/importing DCC charts.

XML export at project level

In SCOUT/STARTER, select **Project -> Save and export...** from the menu or **Expert -> Save and export project** from the context menu, and in the **Export Project** dialog that appears, use

the **Browse...** button to specify the target directory for the export or enter this directly in the text field.
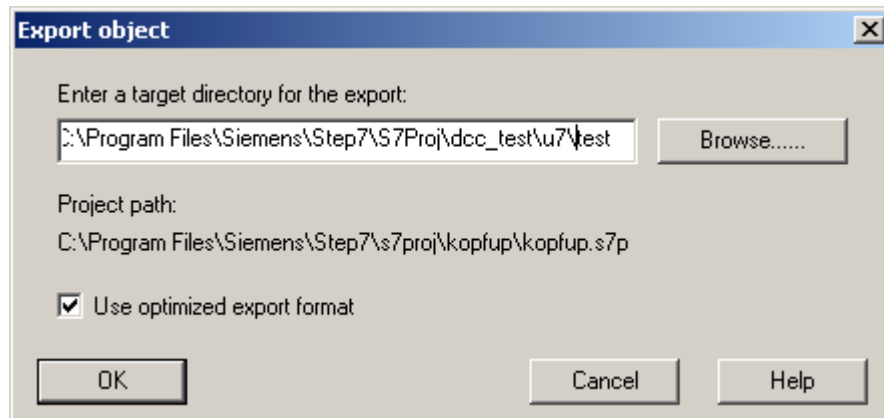
Figure 3-62   Project XML export

Click **OK** to start the export. You can track the progress of the export on the **XML export/import status display** tab.

If the target directory already exists, you will see the following dialog:
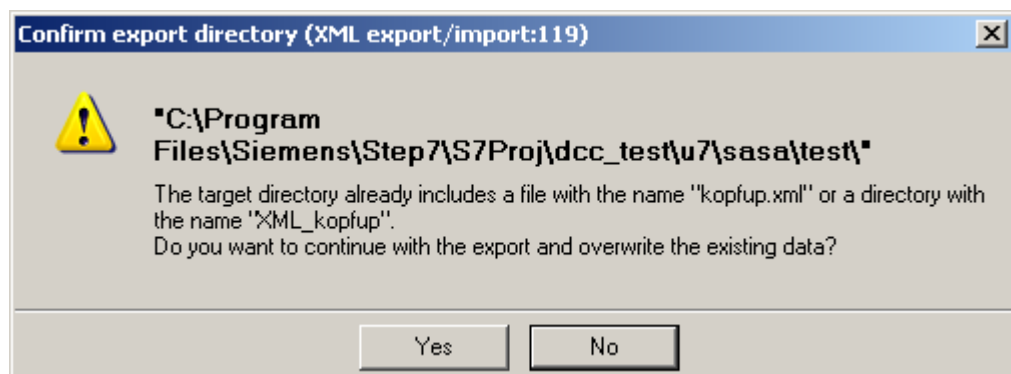
Figure 3-63   XML export - Error message

Choose **Yes** to overwrite the data in the target directory or **No** to cancel the export.

XML export at DCC chart level

Select the DCC chart you wish to export in the project navigator, then select **Expert -> Save project and export object** from the context menu in SCOUT/STARTER. In the following **Export object** dialog, use the **Browse...** button to specify the target directory for the export or enter this directly in the text field.

Click **OK** to start the export. You can track the progress of the export on the **XML export/import status display** tab.

Note

DCC charts that were exported from SCOUT/STARTER cannot be imported into the SIMATIC Manager.

XML import

DCC charts and DCC libraries can be individually exported from SCOUT/STARTER and reimported into a SCOUT/STARTER project. They have the same layout after the import as they did before the export. No information is lost from the DCC chart, i.e. the subcharts, library information, sheet bar interconnections, execution groups, alias definitions, p21000 settings, and know-how protection are all retained during an XML export/import. Know-how protection can be revoked after the import, making it possible to edit the charts again.

You also have the option of importing a previously exported DCC chart into an existing chart.

When importing DCC charts, a distinction is made between importing a DCC chart and importing into an existing DCC chart:

● Container in project navigator -> Expert -> Import object

● DCC chart -> Export/import object

For an XML import, proceed as follows:

In the project navigator, select the object into which you wish to import, and then select **Expert -> Import object** from the context menu in SCOUT/STARTER.

In the dialog that appears next (**Import object**), use the **Browse...** button to specify the source directory for the import or enter this directly in the text field.
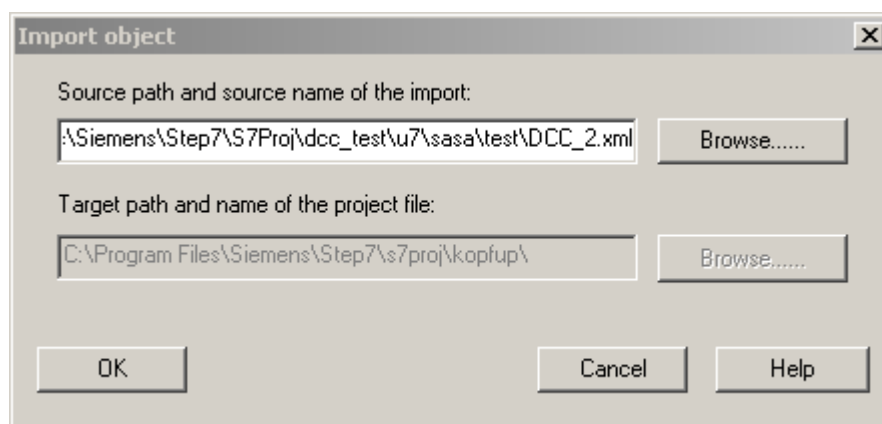

Figure 3-64   XML import

Click **OK** to close the dialog.

In the **Import object** dialog shown below, use the **OK** button to start the import or the **Cancel** button to cancel the import.
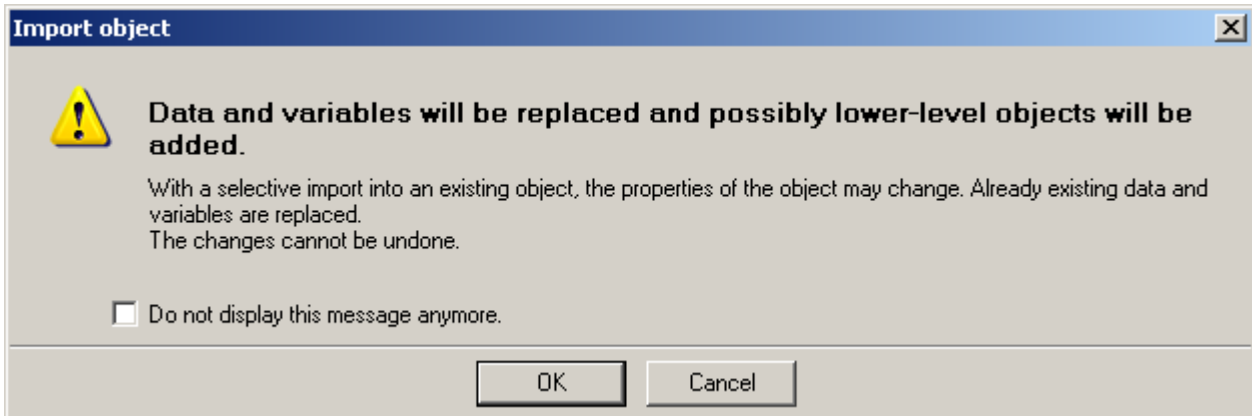
Figure 3-65   XML import warning

You can track the progress of the import on the **XML export/import status display** tab.

Note

It is possible to export/import DCC charts even if the chart in the current project has not been compiled or cannot be compiled without errors.

If the DCC chart has not been compiled so that it is up to date before the export/import, then the imported DCC chart will be empty or will not be up to date.

Even in this case, DCC charts can still be imported without any information being lost, provided that the libraries used in the chart are not yet installed or were imported on the program container. Reinstall the SIMOTION libraries by selecting **Options -> Install libraries...**, or by selecting **Select technology packages...** from the context menu in the case of SINAMICS.

The XML export/import can be carried out for DCC charts with existing DCC chart sources or for DCC charts without DCC chart sources. Please observe the following:

● During the import of DCC charts with DCC chart sources, the set values of the expert list are overwritten with the set values and BICO parameters from the DCC project.

● During the import of DCC charts without DCC chart sources, the adjustable parameters and BICO parameters (@ parameters) are not overwritten.

If the DCC chart sources are not available in a project because the DCC charts were obtained as a result of an upload from the target device or copying without a DCC license, or the DCC chart sources were explicitly deleted, the DCC chart is exported and imported without chart sources. The DCC chart that is created in this way can be loaded and compiled. You can select **Block types** from the context menu to exchange the version of the subordinate basic DCB libraries.

See also

Exchanging the basic library version for installed libraries (Page 83)

3.18 XML export/import of DCC libraries

DCC libraries can be exported/imported in their entirety. Alternatively, it is possible to perform an XML export of individual DCC blocks from the DCC library and an XML import into a DCC library.

Data that is specific to the library (description of block connections, comments, block family, etc.) is exported/imported along with it.

Before the library sources are deleted, a check is performed to determine whether the libraries have been compiled so that they are up to date. As of SCOUT/STARTER 4.2, DCC chart sources are always exported/imported too if they are available on the library sources.

3.19 Reading back DCC chart sources from the target device

In DCC 2.0, DCC charts can be read back from the target device and loaded to another target device of the same type. To enable the DCC charts to be edited further, the DCC chart sources must be present in the original project.

As of DCC 2.1, DCC charts for which no up-to-date DCC chart sources or no DCC chart sources at all are present can be uploaded from the target device and read back to the DCC editor.

Chart sources for DCC libraries (typicals) are downloaded/uploaded via the library context menu. They are uploaded as part of the project uploading process.

Procedure

Under **Options -> Settings** in SCOUT, select the **Download** tab and activate the **Include DCC chart data** checkbox below the **Store additional data on the target device** checkbox to load the DCC chart sources to the target device.
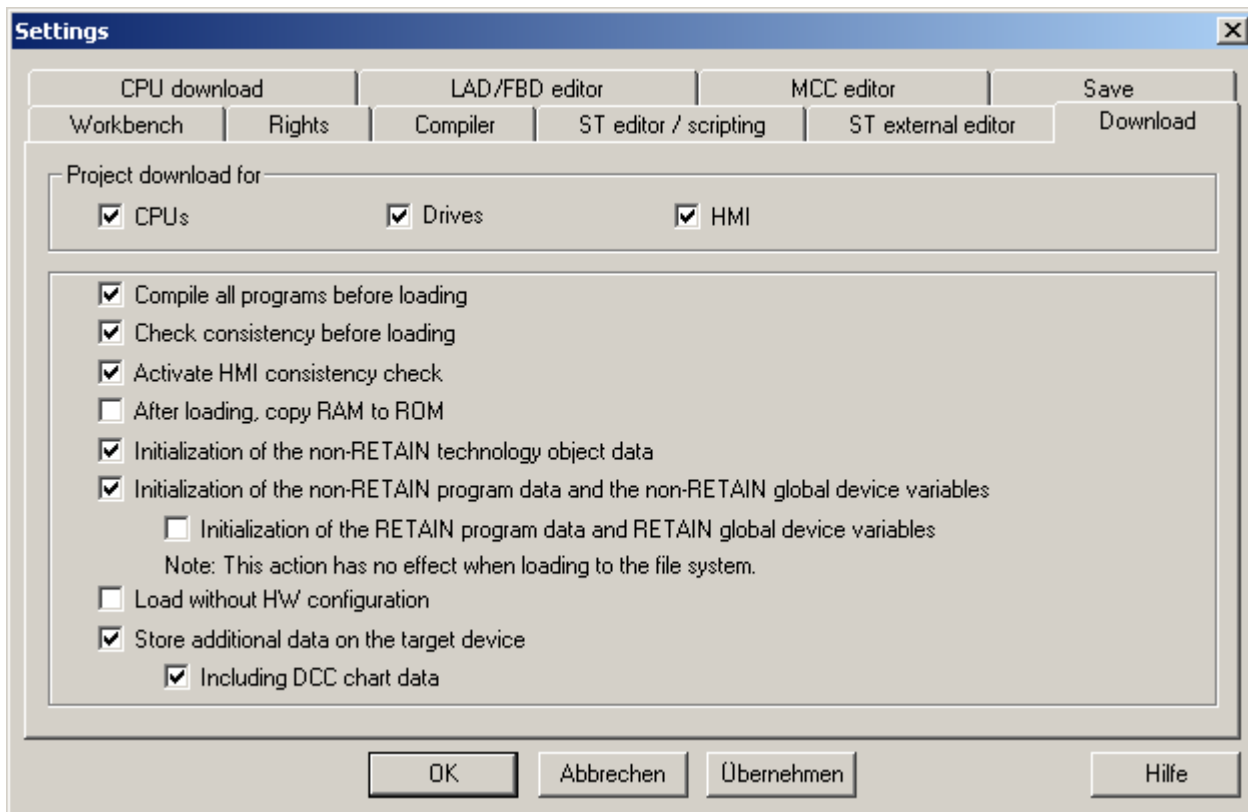
Figure 3-66   Storing additional data and sources on the target device - SCOUT

In STARTER, under **Options -> Settings,** select the **Download** tab and activate the **Store additional data on the target device** checkbox to download the DCC chart sources to the target device.
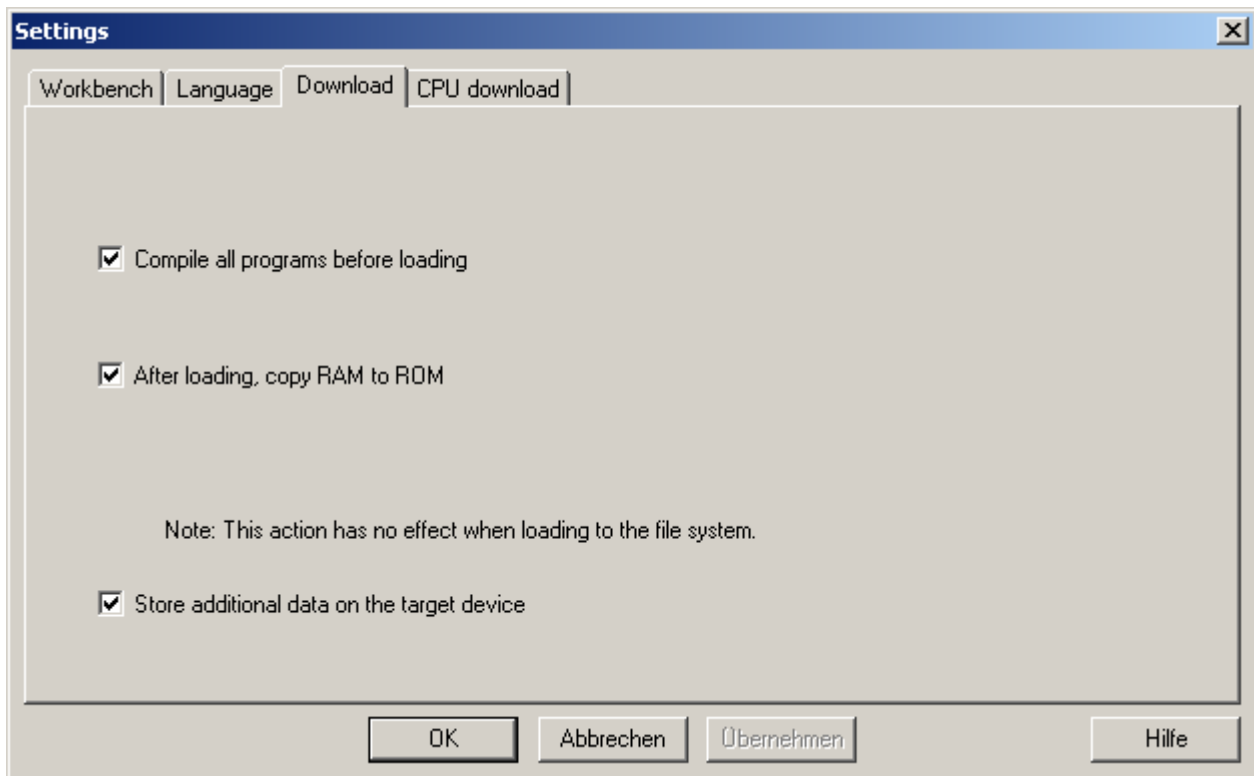
Figure 3-67   Storing additional data and sources on the target device - STARTER

During the CPU download, you can also select whether the DCC chart sources are to be downloaded to the target device for the CPU.

The default setting for the CPU-specific option is taken from **Options -> Settings -> Download**. The option selected for the CPU is then saved as the default setting for the next download.

Once you change the option under **Options -> Settings -> Download,** this is regarded as the default setting for all CPUs.
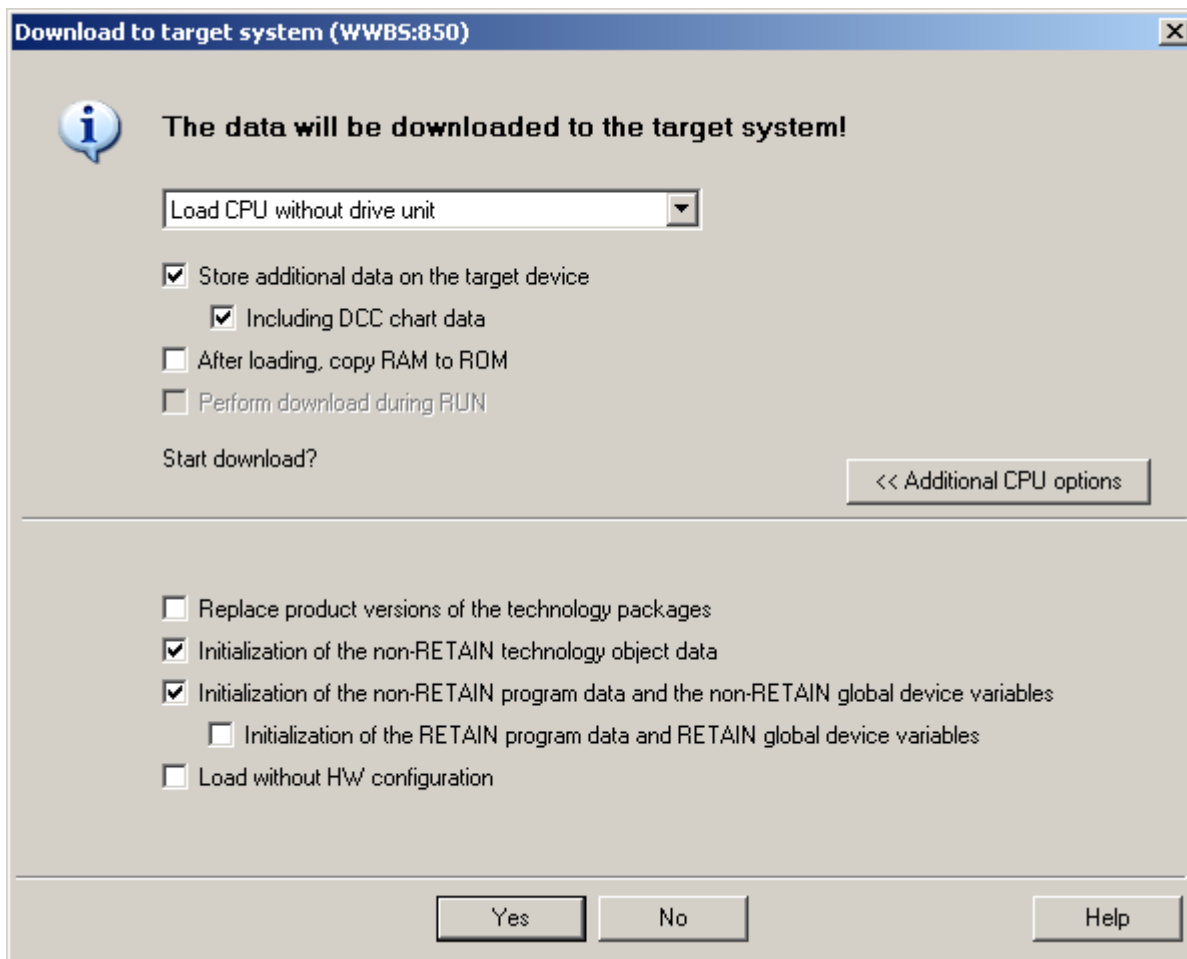
Figure 3-68   Downloading to the target device - SCOUT
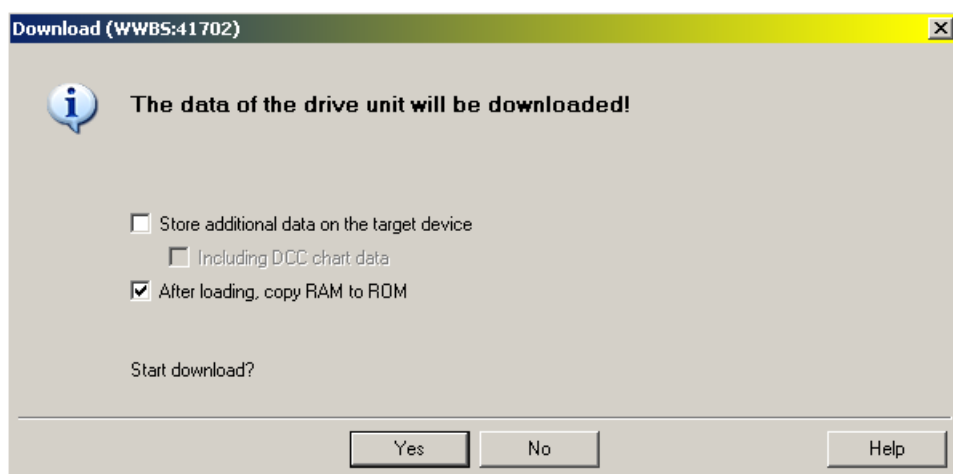
Figure 3-69   Downloading to the target device - STARTER

If you deactivate the **Store additional data on the target device** option, the chart sources in the target device are deleted during the next download.

If you activate the **Store additional data on the target device** option, the chart sources of the DCC libraries are downloaded to the target device, which are used in the DCC charts of the CU or CPU.

If DCC charts are available on the target device, the project is saved following upload.

When downloading to the PG, you must explicitly select whether the library sources from the target device will be loaded into the offline project. In this way, you can keep different library versions in different devices.
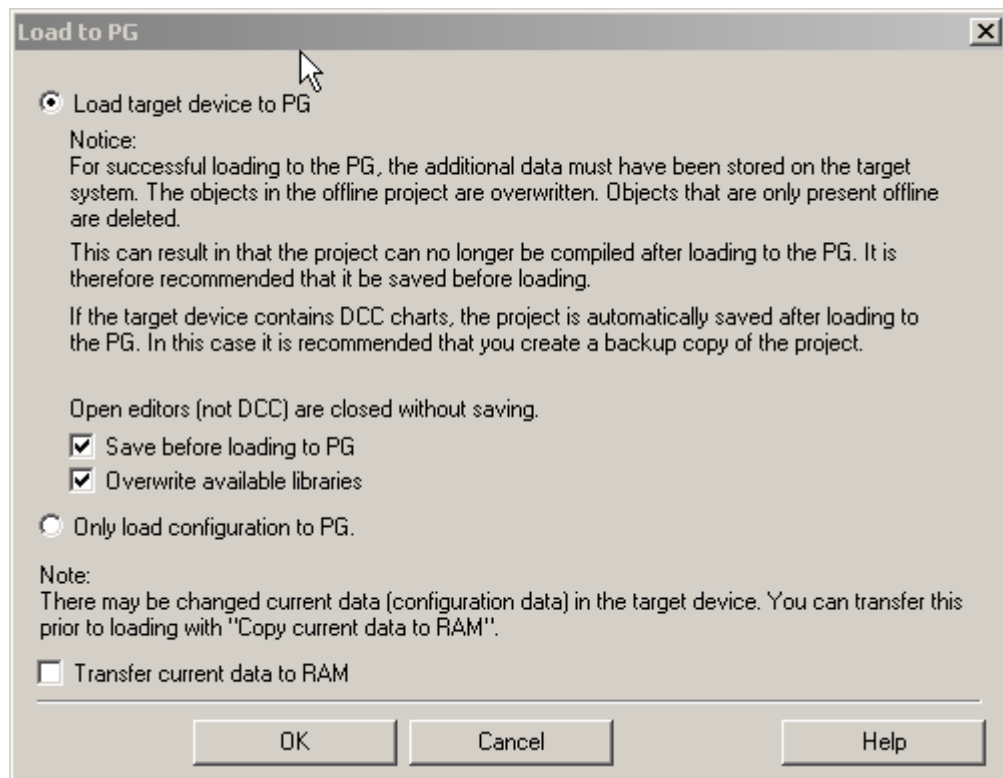
Figure 3-70   Downloading target device to PG - SCOUT
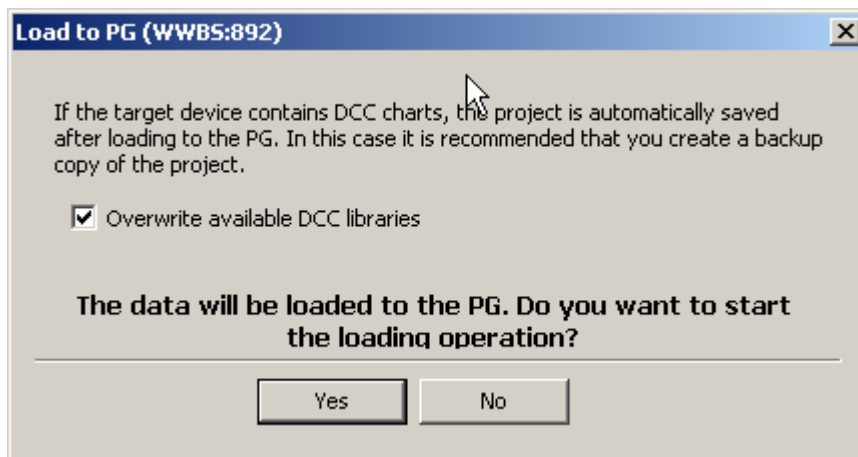
Figure 3-71   Downloading target device to PG - STARTER

Note

In the project comparison, DCC charts are displayed differently following read-back (incl. DCC chart sources), as the DCC chart sources are recreated and therefore have a different time stamp.

Conditions

- If the DCC chart is not open, the chart source will be updated automatically when the project is read back.

- If DCC charts are open in the editor during the project upload, the chart sources will not be updated automatically in the editor. In editor mode, the chart that has been read back can be updated from the CFC using function key **F5**.

- The chart must not be in test mode while it is being read back.

- If the chart sources are not present in SCOUT/STARTER, the DCC chart sources will be created automatically after the project has been uploaded.

- During the project upload, DCC charts with the same name that are already present in the engineering system will be overwritten.

- If the chart names are the same, DCC sources with know-how protection overwrite existing chart sources during the upload to the engineering system.

- DCC libraries are restored when the DCC chart sources are read back.

- The layout of the charts after the read-back process is the same as when they were edited: The positions of the block instances are restored and comments are available in the same position as they were before.

- The libraries used in the charts must be installed in SIMOTION SCOUT/STARTER so that the upload of DCC charts is performed without errors.

The process of reading back DCC charts is independent of the SIMOTION firmware version.

For SINAMICS, it is possible as of SINAMICS V4.4.

The additional data for the read-back process must have been loaded to the target device. Additional data for DCC charts can be created and loaded as of DCC 2.1.

When charts with know-how protection are read back, the know-how protection for the DCC charts is restored in the engineering system.

Changes that have been made online in DCC test mode are taken into account when DCC chart sources are read back.

The following restriction applies if you do not have a DCC license / the CFC has not been installed:

- DCC charts can be uploaded and loaded back onto the target device, but they can no longer be edited - even if the CFC is installed or a DCC license is made available after this.

Note

Up to STARTER/SCOUT V4.1.5, the process of reading back BICOs/parameters for DCC SINAMICS had to be explicitly initiated from the editor after the project had been read back from the target device.

With STARTER/SCOUT V4.2, however, the BICOs/parameters are automatically read back when the project is read back into the DCC chart.

DCC for SINAMICS

4

4.1 Overview

4.1.1 Introduction

This product brief is intended for first-time users who are not yet familiar with the DCC program package. Using a short example, you will find out how to create a new DCC, interconnect DCBs (drive control blocks; function blocks), compile the chart, download it to the target device and test it.

Note

In most cases, there are a number of options for working with the DCC editor (e.g. using the keyboard). In this example, one option is used. Apart from a few exceptions, no alternative methods of operation are explored here.

Note

The documentation on SINAMICS refers to versions STARTER 4.4, DCC 2.3 and SINAMICS Firmware 4.7.

4.1.2 Software requirements

The software requirements for DCC are the same as those for SINAMICS STARTER.

You will additionally need:

- **DCC SINAMICS** software option package

Note

The DCC SINAMICS option package contains on a USB stick the Step7 CFC license required for the DCC editor. The installation is made using the Step7 Automation License Manager (refer to the help of the Step7 Automation License Manager). The Step7 Automation License Manager program is installed automatically together with STARTER.

When the product is supplied, the DCC technology option is not yet available on the drive unit. The DCC technology option must be loaded onto the drive unit's CF card in a separate operation using SIMOTION SCOUT or STARTER. Afterwards, the SINAMICS CU3x0, SINAMICS DC MASTER or the SIMOTION D4xx must be switched off and then on again. Only then can the DCCs be downloaded and run on the drive objects.

From SINAMICS 4.3, the dcblib for S120 is already present on the card. For single drive units, POWER ON after the download of the dcblib is no longer necessary.

There is only ever one DCC on a drive object. DCC can be activated simultaneously on several drive objects on a drive unit. DCC is not available on SINAMICS S110 drive units (CU305 module).

4.1.3 SINAMICS system integration

4.1.3.1 Applications and features

Application

A logic operation, which connects several states (e.g. access control, plant status) to a control signal (e.g. ON command), is required for controlling the drive system in a wide variety of applications.

As well as logic operations, a number of arithmetical operations / storing elements are increasingly becoming a requirement in drive systems.

Note

This additional functionality increases the computing time load. This means that the maximum possible configuration for a Control Unit is restricted.

Drive Control Chart (DCC) functionality is available for every drive object of the drive unit listed in the following table.

Table 4-1   Drive object types for DCC SINAMICS

Drive object type	Object number (r0107)	Meaning
CU_S	1	Control Unit SINAMICS S (SINAMICS S120/S150)
CU_G	2	Control Unit SINAMICS G (SINAMICS G130/G150)
CU_I	3	Control Unit SINAMICS Integrated
CU_CX32	4	Controller Extension for boosting the computing performance
CU_GM	5	Control Unit SINAMICS GM
CU_DC	6	SINAMICS DC MASTER Control Unit
CU_GL	7	Control Unit SINAMICS GL
CU_SL	101	Control Unit SINAMICS SL
A_INF	10	Active Infeed control
SERVO	11	Servo control
VECTOR	12	Vector control
VECTORMV	13	Vector control for SINAMICS GM
VECTORGL	14	Vector control for SINAMICS GL
VECTORSL	16	Vector control for SINAMICS SL

Drive object type	Object number (r0107)	Meaning
DC_CTRL	17	Closed-loop control for DC drives
S_INF	20	Smart Infeed control
B_INF	30	Basic Infeed control
A_INFMV	40	Active Infeed control for SINAMICS SM150
B_INFMV	41	Basic Infeed control for SINAMICS GM150
TB30	100	Terminal Board 30
TM31	200	Terminal Module 31
TM41	201	Terminal Module 41
TM15DI/DO	204	Terminal Module 15 (for SINAMICS)
TM120	207	Temperature evaluation with safe electrical isolation
DO encoder	300	Drive object encoder
HLA	70	Hydraulic Linear Axis
RIC	21	Renewable Infeed Control

4.1.3.2 Execution groups in the DCC editor

Description

Exactly one DCC chart can be created per drive object (DO = drive object). This DCC chart can contain up to ten execution groups.

Execution groups in the DCC editor

Execution groups are groups of blocks. The blocks of an execution group are started in a defined sequence at a specified time and are calculated cyclically one after the other within a specified sampling time.

In the DCC editor a maximum of ten execution groups (execution group 1 to 10) can be defined per drive object (DO) and thereby per basic chart (see also: SINAMICS system integration > Execution sequence, creating new execution groups). In the project navigator of STARTER / SCOUT (context menu of the chart, menu command **Set execution groups ...**) fixed or free execution groups of the drive object can be assigned to the previously defined execution groups.

Note

The **Interconnection with execution group**... command is only supported with DCC SIMOTION.

> ⚠ **WARNING**
>
> **Changing the assignment of an execution group**
>
> If the assignment of an execution group is changed in the **Set Execution Groups** window (or in parameter p21000[]), the affected execution group is initially logged off during time slice management and then logged on once again with the new assignment. The execution group is not calculated during the period between logging off and logging back on. The log-on and log-off take place in a background process of the drive unit; the duration is therefore not defined and is determined by the current CPU time load of the drive unit. (This affects the path of the output signal in the case of time-dependent blocks, e.g. the DIF differentiator.) Prior to the first calculation cycle after logging back on, internal status variables of the blocks are partially reset. For both of these reasons, this can result in jumps in the output signal of blocks, which for example can affect the torque/force setpoint and (in the case of operated axes) also the torque/actual force value. Logic signals can also acquire an unexpected state at this point of operation.

If the change to the execution group also results in a change to the sampling time, internal constants or factors are automatically adjusted for time-dependent blocks (BF, DCA, DIF, DT1, INT, MFP, PCL, PDE, PDF, PIC, PST, PT1, RGE, RGJ, WBG). If you use these blocks in the following execution groups, you must assign parameter p2048 the value of the isochronous master cycle clock:

- Receive AFTER IF1 PROFIdrive PZD

- Send BEFORE IF1 PROFIdrive PZD

- Receive AFTER IF1 PROFIdrive flexible PZD

- Receive AFTER IF2 PZD

- Send BEFORE IF2 PZD

- Receive AFTER IF2 flexible PZD

Note that the execution groups created in the chart are only visible in the **Set Execution Groups** window after the chart has been compiled.

Parameter r21002 displays the basic sampling time hardware.

Parameter r21003 displays the basic sampling time software.

For more detailed information on these parameters, see the online help for DCC parameters or the "DCC blocks" Function Manual.
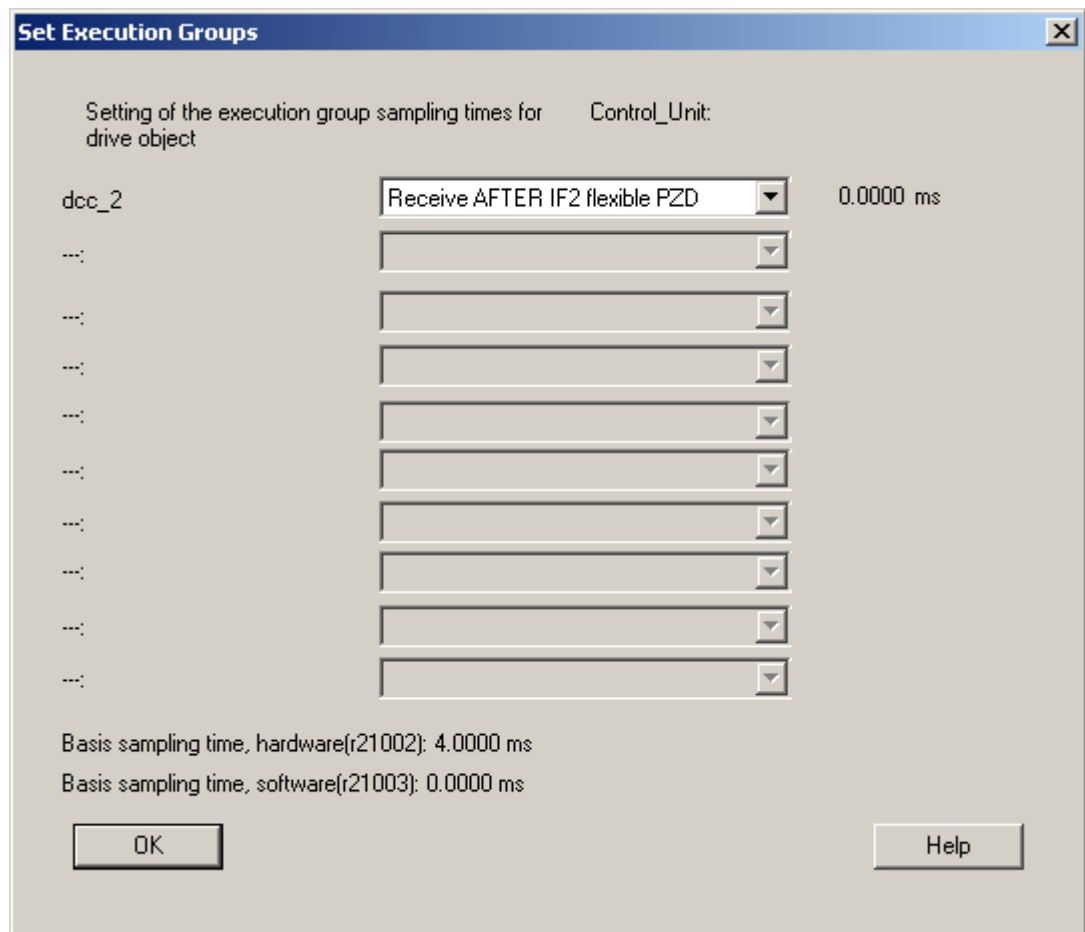
Figure 4-1   **Set Execution Groups** window

The execution groups created in the DCC editor must be assigned to the "Fixed execution groups" and the "Free execution groups" of the SINAMICS drive object.

Note

For parameter p21000 (Properties execution group) and, therefore, in the context menu of the "Set sampling times" chart, write access for the enabled controller is blocked in at least one drive axis or infeed. The old value automatically takes effect again in this case. If necessary, switch off all drive axes and infeeds, and make a new attempt.

4.1.3.3   Fixed execution groups

Fixed execution groups

A "fixed execution group" is called at a fixed position in the system execution using the sampling time of the corresponding system function.

The following fixed execution groups are available for SINAMICS as of FW Version 2.5:

- **Read in AFTER digital inputs**
  This execution group is called after the current values of the digital inputs have been read in on this drive object type and the corresponding binector outputs have been written. The sampling time of this execution group corresponds to the sampling time of the CU inputs/outputs (p0799) or the digital inputs/outputs of the TB30, TM31 and TM41 (p4099[0]).

- **Output BEFORE digital outputs**
  This execution group is called before the digital outputs are output on this drive object type. The sampling time of this execution group corresponds to the sampling time of the CU inputs/outputs (p0799) or the inputs/outputs of the TB30, TM31 and TM41 (p4099[0]).

- **Read in AFTER analog inputs**
  This execution group is called after the current values of the analog inputs have been read in on this drive object type and the corresponding binector outputs have been written. The sampling time of this execution group corresponds to the sampling time of the inputs/ outputs of the TB30, TM31 and TM41 (p4099[1]).

- **Output BEFORE analog outputs**
  This execution group is called before writing is performed on the analog outputs. The sampling time of this execution group corresponds to the sampling time of the inputs/ outputs of the TB30, TM31 and TM41 (p4099[1]).

- **BEFORE speed controller**
  This execution group is called before the speed controller additional setpoint values "n_reg n_set1" (p1155) and "n_reg n_set2" (p1160) are read into FP3080. The sampling time of the speed controller (p0115[1]) produces the call, but a minimum sampling time of 1 ms is required.

- **BEFORE speed setpoint channel**
  This execution group is called before function block diagrams 3010, 3020, 3030, 3040 and subsequent charts are calculated, if the setpoint channel has been activated (p0108.8 = 1). If no setpoint channel has been configured (p0108.8 = 0), calculation takes place before function block diagram 3095. The setpoint channel sampling time produces the call (p0115[3]).

- **BEFORE position controller**
  This execution group is called after the actual position value preparation (function block diagram 4010) has been calculated and before the position controller (function block diagrams 4015, 4020 and 4025) is calculated. The sampling time of this execution group corresponds to the sampling time of the position controller (p0115[4]).

- **BEFORE actual position value**
  This execution group is called before the actual position value preparation (function block diagram 4010) and the position controller (function block diagrams 4015, 4020, and 4025) are calculated. The sampling time of this execution group corresponds to the sampling time of the position controller (p0115[4]). (Available as of V4.3).

- **BEFORE basic positioner**
  This execution group is called before the basic positioner function module (function block diagrams 3610 to 3650) is calculated. The sampling time of this execution group corresponds to the sampling time of the basic positioner function module (p0115[5]).

- **BEFORE standard technology controller**
  This execution group is called before the technology controller function module is calculated (p0108.16 = 1) (function block diagrams 7950, 7954 and 7958). The sampling time of this execution group corresponds to the sampling time of the technology controller (p0115[6]).

- **Receive AFTER IF1 PROFIdrive PZD** [1]
  This execution group is called after cyclic process data has been received via communication interface IF1 (e.g. via the integrated PROFIBUS interface) and output to connector outputs r2050[..], r2060[..], binector outputs r2090 - r2093 and connector binector transformers r2094 and r2095.
  The sampling time of this execution group corresponds to the PROFIdrive PZD sampling time. As of SINAMICS FW Version 2.5, communication interface IF1 (Interface 1) is always used by the PROFIBUS interface inside the CU or, if a CBE20 has been inserted in the option slot, PROFINET.
  Parameter p0092 can be used to set whether the utilization calculation (r9976) evaluates the execution group for isochronous operation (p0092 = 1) or non-isochronous operation (p0092 = 0) during SINAMICS ramp-up.

  Non-isochronous operation (r2043.1 = 0):
  For SINAMICS **software versions V2.5 and V2.6**, the PROFIBUS data is sent and received immediately after each other at the start of the IF1 PROFIdrive PZD sampling time.
  At the start of the IF1 PROFIdrive PZD sampling time set in p2048, the PROFIBUS data is sent first and then the process data received. In the case of non-isochronous cyclic communication at the IF1 communication interface, the execution group is cyclically calculated with the sampling time of the IF1 communication interface (p2048), i.e. after writing the connector outputs for the process data (PZD) r2050[..], r2060[..], r2090 - r2093 and after calculating the connector binector transformer r2094 and r2095.
  As of SINAMICS **software version V4.3**, the PROFIBUS data is received and sent at two separate times. The process data is received at the start of the IF1 PROFIdrive PZD sampling time set in p2048. In the case of non-isochronous cyclic communication at the IF1 communication interface, the execution group is cyclically calculated with the sampling time of the IF1 communication interface (p2048) after receiving the data, i.e. after writing the connector outputs for the process data (PZD) r2050[..], r2060[..], r2090 - r2093 and after calculating the connector binector transformer r2094 and r2095.
  For calculating the utilization of system r9976 correctly, p092 must be = 0 (== factory setting) in non-isochronous operation.

  Isochronous operation (r2043.1 = 1):
  In isochronous operation, the times at which the data is received from the master (T_o), the data is sent to the master (T_i) and the DP cycle time (T_{DP}) [3] are configured in the master. Internally, the calls to T_o and T_i are realized using a state machine that is called cyclically with sampling time T_{Zu}, e.g. for standard servo drive objects with the current controller sampling time of 125 µs. The sampling time T_{Zu} = 125 µs, however, is at least as large as the largest current controller sampling time (e.g. for the vector, 250 µs, 375 µs or 500 µs). This means only integer multiples of $T_{Zu} >= 125$ µs can be set [3] as times T_i and T_o for the drive.
  $\rightarrow T_i = n_i \cdot T_{Zu}$:

Figure 4-2   Calling the "Receive AFTER IF1 PROFIdrive PZD" or "Receive AFTER IF2 PROFIdrive PZD" execution group for isochronous PROFIBUS communication. See also [3].

Receive data (sent by the master) is always processed and made available at the end of the sampling time $(n_o - 1) T_{Zu}$, so that the received process data remains valid until the start of the next sampling period at the time $T_o = n_o \cdot T_{Zu}$ at r2050[..], r2060[..], r2090 - r2093. Thus, in the last sampling time T_{Zu} before T_o. The "Receive AFTER IF1 PROFIdrive PZD" execution group is then calculated in the following first sampling time T_{Zu} after T_o.

Note

Please take note that two connector-binector converters (function block diagram 2468: p2099, r2094, r2095) are NOT called synchronously in isochronous mode but may be processed within sampling time p2048 at any time depending on the computing time utilization.

The calculation must be completed within the sampling time T_{Zu}; otherwise, warning A01053 "System overload detected" or a time-slice overflow (F01205) will occur. For this reason, only the absolute minimum number of DCBs that are necessary for the required function should be calculated in this execution group.

The computing time available for this execution group decreases the more drive axes are calculated on the CU since the current controllers (and also the speed controller for the servo) of the drive axes are also calculated in the sampling time $p0115[0] = T_{Zu}$ (>=125 µs).

If the utilization calculation is to be executed during ramp-up of the CU for isochronous operation (unfavorable case regarding degree of utilization), p092 = 1 must be set. Otherwise (p092 = 0) the utilization in r9976 will be updated only after switchover to isochronous operation. The increased maximum overall utilization in isochronous operation will be displayed only in r9976.

If you want to calculate a larger number of DCBs, please check whether you would not do better to configure this using the "Receive AFTER IF1 PROFIdr. flexible PZD" execution group.

Note

In the case of this execution group, please note that the higher computing time utilization that applies in isochronous operation is not actually taken into account in the utilization calculation until the time of the transition (specified by the PROFIBUS master) to isochronous operation. After the drive device has ramped up, this means the utilization of the complete system initially still lies within the valid range and it is only when the transition is made to isochronous operation that the drive device shuts down with fault F1054 "System limits exceeded".

- **Send BEFORE IF1 PROFIdrive PZD** [1)]
  This execution group is called before cyclic process data is sent via communication interface IF1 (e.g. the integrated PROFIBUS interface), i.e. before calculating the connector binector transformer p2080 – p2084 and reading in the connector inputs p2051[..], p2061[..]. The sampling time of this execution group corresponds to the PROFIdrive PZD sampling time. As of SINAMICS firmware Version 2.5, communication interface IF1 is always used by the PROFIBUS interface inside the CU or, if a CBE20 is inserted in the option slot, PROFINET.
  Parameter p0092 can be used to set whether the utilization calculation (r9976) evaluates the execution group for isochronous operation (p0092 = 1) or non-isochronous operation (p0092 = 0) during SINAMICS ramp-up.

  Non-isochronous operation (r2043.1 = 0):
  For SINAMICS software versions **V2.5** and **V2.6**, the PROFIBUS data is sent and received (in this sequence) immediately after each other at the start of the IF1 PROFIdrive PZD sampling time.
  The "Send BEFORE IF1 PROFIdrive PZD" execution group is processed at the end of the IF1 PROFIdrive PZD sampling time before the data is sent at the start of the next sampling time. After computing the execution group, at the end of the sampling time, the connector binector transformer p2080 – p2084 is still calculated and the connector inputs p2051[..], p2061[..] read in. This means all send data is present completely at the end of the sampling time and will be sent at the start of the next sampling time.

  As of SINAMICS software version **V4.3**, the PROFIBUS data is received and sent at two separate times.
  The "Send BEFORE IF1 PROFIdrive PZD" execution group is processed at the end of the IF1 PROFIdrive PZD sampling time before the complete data is still sent at the end of the sampling time. For the non-isochronous (cyclical) communication at the communications interface IF1, the execution group is calculated cyclically with the sampling time of the interface IF1 (p2048) (before sending the data), i.e. also before calculating the connector binector transformer p2080 – p2084 and reading in the connector inputs p2051[..], p2061[..].

For calculating the utilization of system r9976 correctly, p092 must be = 0 (== factory setting) in non-isochronous operation.

Isochronous operation (r2043.1 = 1):

In isochronous operation, the times at which the data is received from the master (T_o), and sent to the master (T_i) as well as the DP cycle time (T_{DP}) [3] are configured in the master. Internally, the calls to T_o and T_i are realized using a state machine that is called cyclically with sampling time T_{Zu}, e.g. for standard servo drive objects with the current controller sampling time of 125 µs. The sampling time T_{Zu} = 125 µs, however, is at least as large as the largest current controller sampling time (e.g. for the vector 250 µs, 375 µs or 500 µs). This means only integer multiples of T_{Zu} >= 125 µs can be set [3] as times T_i and T_o for the drive.

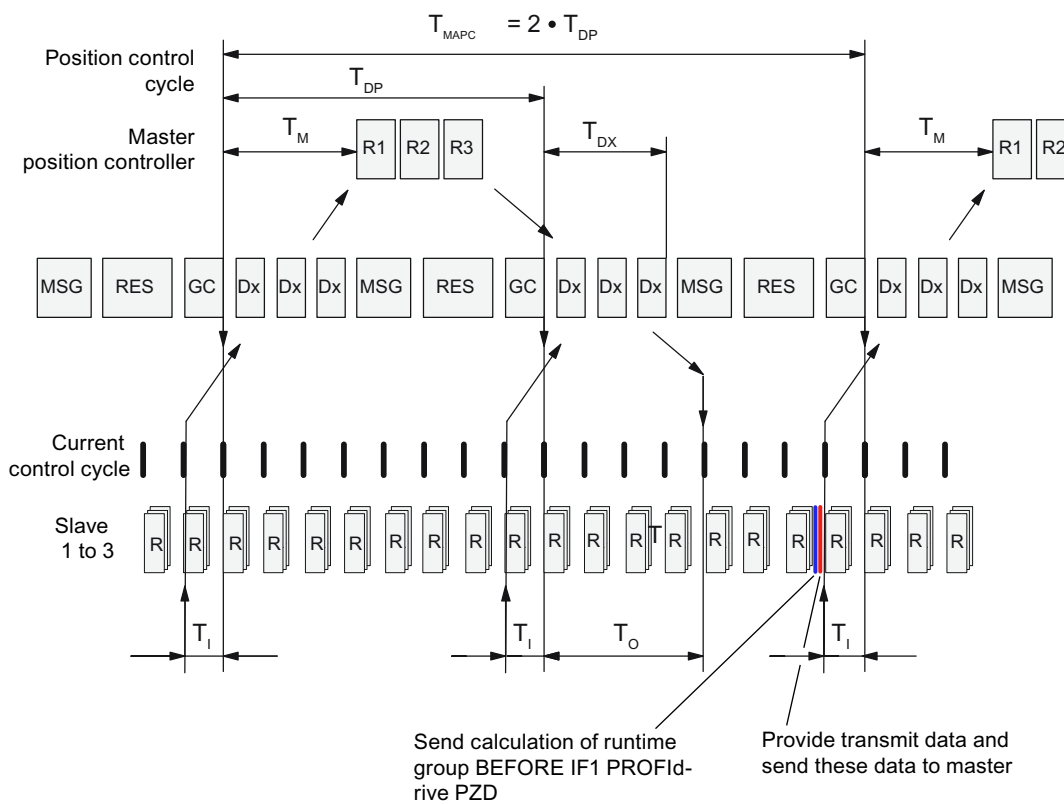$$\rightarrow T_i = n_i \cdot T_{Zu} \text{ and } T_{DP} = n_{DP} * T_{Zu}$$

Figure 4-3   Calling the "Send BEFORE IF1 PROFIdrive PZD" or "Send BEFORE IF2 PROFIdrive PZD" execution group for isochronous PROFIBUS communication. See also [3].

The "Send BEFORE IF1 PROFIdrive PZD" execution group is calculated in the last sampling time T_{Zu} before the data is sent to the master and before T_i and reading in the connector inputs p2051[..], p2061[..].

Note

Please note that five connector-binector converters (function block diagram 2472: p2080 ..., r2089) are also NOT called synchronously in isochronous operation but may be processed within sampling time p2048 at any time depending on the computing time utilization.

The calculation of the execution group must be completed within sampling time T_{zu}]; otherwise, a fault F1054 "System limit exceeded" (V4.3), a warning A01053 "System overload measured" (V2.x) or a time-slice overflow (F01205) occurs. For this reason, only the absolute minimum number of DCBs that are necessary for the required function should be calculated in this execution group.
The computing time available for this execution group decreases as the number of drive axes calculated on the CU increases, since the current controller (and the speed controllers for the servo) of the drive axes are also calculated in sampling time $p0115[0] = T_{zu}$.

If the utilization calculation during the ramp-up of the CU has already executed the isochronous operation (the more unfavorable configuration regarding the utilization of the CU), then p092 must be set to = 1. Otherwise (p092 = 0) the utilization in r9976 will be updated only after switchover to isochronous operation. The increased maximum overall utilization in isochronous operation will be displayed only in r9976.

Note

In the case of this execution group, please note that the higher computing time utilization that applies in isochronous operation is not actually taken into account in the utilization calculation until the time of the transition (specified by the PROFIBUS master) to isochronous operation. After the drive device has ramped up, this means the utilization of the complete system initially still lies within the valid range and it is only when the transition is made to isochronous operation that the drive device shuts down with fault F1054 "System limits exceeded".

- **Receive AFTER IF1 PROFIdr. flexible PZD** [1]

  This execution group is called after cyclic process data has been received via communication interface IF1 (e.g. via the integrated PROFIBUS interface) and output to connector outputs r2050[..], r2060[..], binector outputs r2090 - r2093 and connector binector transformers r2094 and r2095.

  The sampling time of this execution group corresponds to the PROFIdrive PZD sampling time. The only difference from the Receive AFTER IF1 PROFIdrive PZD execution group is the way in which the execution group behaves in isochronous mode. Even in isochronous mode this execution group is called using the PROFIdrive PZD sampling time configured in the master as with any other sampling time. This means that initially all shorter sampling times are called at time T_o depending on the validity of the receive data (current controller; speed controller if applicable). This execution group is called first only when the T_{DP} sampling time starts to be processed. You basically do not know how often the shorter sampling times with higher priority of the current and speed controller are calculated before this execution group starts to be processed. In addition, processing of this execution group is interrupted due to shorter sampling times.

  This execution group should only be used if IF1 is usually operated in isochronous mode. The benefit of this execution group is that considerably more blocks can be calculated than in the Receive AFTER IF1 PROFIdrive PZD execution group because the calculation does not have to be completed after current controller sampling time p0115[0]. However (due to the higher priority and interruption caused by shorter sampling times), there is no longer a fixed synchronism between T_o and the time the execution group is called.

- **Receive AFTER IF2 PZD** [2)]
  The CAN bus interconnection CBC10, the CBE20, or the integrated PROFIBUS/
  PROFINET interface (see description p8839, p8815) on the CU can all serve as
  communications interface IF2. This execution group is available only on the CU_S, CU_G,
  CU_GM, CU_GL, SERVO, VECTOR, VECTORMV, VECTORSL and VECTOR_GL drive
  object types.

  Non-isochronous operation (PROFINET, PROFIBUS, or CAN bus):
  The execution group is called once the connectors r8850[..], r8860[..], r8890 - r8893 have
  been described with the receive data and the calculation of the connector-binector
  converters (function block diagram 2485.7: r8894 and r8895) has been executed. The
  sampling time of this execution group corresponds to the IF2 PZD sampling time in p8848.

  Isochronous operation (r2043.1 = 1):
  As of SINAMICS V4.4, isochronous PROFINET/PROFIBUS operation is also supported for
  the communications interface IF2. (See SINAMICS S120 Function Manual FH1 →
  Applications → Parallel operation of communications interfaces.)
  In isochronous operation, the times at which the data is received from the master (T_o), the
  data is sent to the master (T_i) and the DP cycle time (T_{DP}) [3)] are configured in the master.
  Internally, the calls to T_o and T_i are realized using a state machine that is called cyclically
  with sampling time T_{Zu}, e.g. for standard servo drive objects with the current controller
  sampling time of 125 µs. The sampling time T_{Zu} = 125 µs, however, is at least as large as
  the largest current controller sampling time (e.g. for the vector 250 µs, 375 µs, or 500 µs).
  This means only integer multiples of T_{Zu} >= 125 µs can be set as times T_i and T_o for the
  drive [3)].
  → $T_i = n_i \cdot T_{Zu}$ and $T_{DP} = n_{DP} * T_{Zu}$ See figure 3-2
  Receive data (sent by the master) is always processed and made available at the end of
  the sampling time (n_o -1) T_{Zu}, so that the received process data remains valid until the start
  of the next sampling period at the time $T_o = n_o \cdot T_{Zu}$ at r8850[..], r8860[..], r8890 - r8893
  (function block diagram 2485). Thus, in the last sampling time T_{Zu} before T_o. The "Receive
  AFTER IF2 PZD" execution group is then calculated in the first sampling time T_{Zu} to follow
  T_o.

Note

Please take note that two connector-binector converters (function block diagram 2485:
p8899, r8894, r8895) are NOT called synchronously in isochronous mode but may be
processed within sampling time p8848 at any time depending on the computing time
utilization.

The calculation must be completed within the sampling time T_{Zu}; otherwise, warning A01053
"System overload detected" or a time-slice overflow (F01205) will occur. For this reason,
only the absolute minimum number of DCBs that are necessary for the required function
should be calculated in this execution group. The computing time available for this execution
group decreases as the number of drive axes calculated on the CU increases, since the
current controllers (and also the speed controllers for the servo) of the drive axes are also
calculated in the sampling time p0115[0] = T_{Zu} (>=125 µs). If the utilization calculation is to
be executed during ramp-up of the CU for isochronous operation (unfavorable case
regarding degree of utilization), p092 = 1 must be set. Otherwise (p092 = 0) the utilization
in r9976 will be updated only after switchover to isochronous operation. The increased
maximum overall utilization in isochronous operation will be displayed only in r9976. If you

want to calculate a larger number of DCBs, please check whether you would not do better to configure this using the "Receive AFTER IF2 PROFIdr. flexible PZD" execution group.

Note

In the case of this execution group, please note that the higher computing time utilization that applies in isochronous operation is not actually taken into account in the utilization calculation until the time of the transition (specified by the PROFIBUS master) to isochronous operation. After the drive device has ramped up, this means the utilization of the complete system initially still lies within the valid range and it is only when the transition is made to isochronous operation that the drive device shuts down with fault F1054 "System limits exceeded".

- **Send BEFORE IF2 PZD** [2]
  The CAN bus interconnection CBC10, the CBE20, or the integrated PROFIBUS/ PROFINET interface (see description p8839, p8815) on the CU can all serve as communications interface IF2. This execution group is available only on the CU_S, CU_G, CU_GM, CU_GL, SERVO, VECTOR, VECTORMV, VECTORSL and VECTOR_GL drive object types.

  Non-isochronous operation (PROFINET, PROFIBUS, or CAN bus):
  The execution group is called before the binector connector transformers p8880 – p8884 are calculated and the before connector inputs p8851[..], p8861[..] with the send data are read. The sampling time of this execution group corresponds to the IF2 PZD sampling time in p8848. See function block diagrams 2487 and 2493.

  Isochronous operation (r2043.1 = 1):
  As of SINAMICS V4.4, isochronous PROFINET/PROFIBUS operation is also supported for the communications interface IF2.
  (See SINAMICS S120 Function Manual FH1 → Applications → Parallel operation of communications interfaces.)
  In isochronous operation, the times at which the data is received from the master (T_o), and sent to the master (T_i) as well as the DP cycle time (T_{DP}) [3] are configured in the master. Internally, the calls to T_o and T_i are realized using a state machine that is called cyclically with sampling time T_{Zu}, e.g. for standard servo drive objects with the current controller sampling time of 125 µs. The sampling time T_{Zu} = 125 µs, however, is at least as large as the largest current controller sampling time (e.g. for the vector 250 µs, 375 µs, or 500 µs). This means only integer multiples of T_{Zu} >= 125 µs can be set as times T_i and T_o for the drive [3].

  → $T_i = n_i \cdot T_{Zu}$ and $T_{DP} = n_{DP} * T_{Zu}$ See also figure 3-3

  The "Send BEFORE IF2 PZD" execution group is calculated in the last sampling time T_{Zu} before the data is sent to the master and before T_i and reading in the connector inputs p8851[..], p8861[..].

Note

Please note that five connector-binector converters (function block diagram 2489: p8880 ..., r8889) are NOT called synchronously in isochronous mode but may be processed within sampling time p8848 at any time depending on the computing time utilization.

The calculation of the execution group must be completed within sampling time T_{Zu}]; otherwise, a fault F1054 "System limit exceeded" (V4.3), a warning A01053 "System overload measured" (V2.x) or a time-slice overflow (F01205) occurs. For this reason, only the absolute minimum number of DCBs that are necessary for the required function should be calculated in this execution group.

The computing time available for this execution group decreases as the number of drive axes calculated on the CU increases, since the current controller (and the speed controllers for the servo) of the drive axes are also calculated in sampling time p0115[0] = T_{Zu}.

- **Receive AFTER IF2 flexible PZD** [1)]
  This execution group is called after cyclic process data (PZD) has been received via communication interface IF2 (e.g. via the integrated PROFIBUS interface) and output to connector outputs r8850[..], r8860[..], binector outputs r8890 - r8893 and connector binector converters r8894 and r8895.

  The sampling time of this execution group corresponds to the PROFIdrive PZD sampling time. The only difference from the Receive AFTER IF2 PROFIdrive PZD execution group is the way in which the execution group behaves in isochronous mode. Even in isochronous mode this execution group is called using the PROFIdrive PZD sampling time configured in the master as with any other sampling time. This means that initially all shorter sampling times are called at time T_o depending on the validity of the receive data (current controller; speed controller if applicable). This execution group is called first only when the T_{DP} sampling time starts to be processed. You basically do not know how often the shorter sampling times with higher priority of the current and speed controller are calculated before this execution group starts to be processed. In addition, processing of this execution group is interrupted due to shorter sampling times. This execution group should only be used if IF2 is usually operated in isochronous mode. The benefit of this execution group is that considerably more blocks can be calculated than in the Receive AFTER IF1 PROFIdrive PZD execution group because the calculation does not have to be completed after current controller sampling time p0115[0]. However (due to the higher priority and interruption caused by shorter sampling times), there is no longer a fixed synchronism between T_o and the time the execution group is called.

1)   IF1 is the abbreviation for communication interface 1. With SINAMICS V2.5 and SIMOTION V4.1, this refers to the integrated PROFIBUS interface or - if a CBE20 has been inserted in the option slot - the PROFINET interface. IF1 supports the PROFIdrive profile and isochronous operation.

2)   IF2 is the abbreviation for communication interface 2. IF2 of the CU320 and CU320-2 can only be used by the CAN bus if a CBC10 module has been inserted in the option slot and SINAMICS software as of V2.5 is used. The CBE20 or integrated PROFIBUS/ PROFINET interface can be assigned to the IF2 via parameters (p8839, p8815). (See SINAMICS S120 Function Manual FH1 → Applications → Parallel operation of communications interfaces.) As of SINAMICS V4.4, isochronous mode is also supported on IF2.

3)   For T_{DP}, T_i and T_o, the formulas and limit values listed in the Function Manual FH1→ Communication → Communication via PROFIBUS DP → Motion Control with PROFIBUS → "Time settings and Meanings" table apply.

The PROFIBUS functionalities described here also apply to PROFINET.

Not every fixed execution group is available on every drive object type. This means, for instance, that SERVO, VECTOR, VECTORMV... drive object types do not contain any digital or analog inputs; therefore, the fixed execution groups for the digital inputs/outputs and analog inputs/outputs are not available here.

Please ensure that the sampling time for DCC SINAMICS is restricted to below 1 ms. As a property of an execution group, select a fixed execution group in which the sampling time for the assigned system function is < 1 ms. This execution group will thus only be called with the sampling time of 1 ms, and will thereby deviate from the assigned system function. Fault F51004 (see r0947) is set to indicate this deviation. The fault value of the fault (r0949) + 1 gives the number of the execution group where the deviation has occurred.

Example for the automatic limitation of the sampling time:

On the SERVO drive object type, select "BEFORE speed controller" as a fixed execution group. The associated system function is the speed controller.
The sampling time of the speed controller is p0115[1] = 125 µs in the factory setting of p0112 = 3.
In contrast to the sampling time of the speed controller, the sampling time of the DCC execution group is set automatically to 1 ms.
Also refer to the description of parameter p0112 in the SINAMICS S List Manual.

⚠ **WARNING**

In the case of an online change to the sampling time (execution group) the START method of the block is called. Initializations in the START method can result in jumps in the signal path.

4.1.3.4   Free execution groups

Free execution groups

The "free execution groups" are only defined by their sampling time (p21000 = 1 to 256 and p21000 = 1001 to 1096). The sampling times are provided centrally for all drive objects on a SINAMICS drive unit. A sampling time can be used simultaneously by several drive objects

and several DCC execution groups. The sampling times are created in two different ways. The two methods have different limits for the maximum possible number of different sampling times.

● Free execution groups whose sampling times are created in the hardware:
   With the hardware sampling times, in p21000[0...9] each integer multiple of the basic sampling time (read-only in r21002) ranging from 1 * r21002 to 256 * r21002 can be generated with the following limits:

   – Minimum sampling time = 1 ms

   – Maximum sampling time = (r21003 - r21002) < r21003 (approx. 8 ms; on the D410 of the DP cycle).

   – On the D410 drive unit, r21003 always matches the DP cycle (PROFIBUS T_DP cycle clock) or with the send cycle clock for PROFINET. If 1 ms is configured as DP cycle / send cycle for PROFINET, this means no hardware sampling times (but only software sampling times) from DCC are used on this device.

   – The number of hardware sampling times on a drive device is limited. The number of still available hardware sampling times can be read from r7903 (as of SINAMICS V2.6). Further information is contained in Chapter DCC for SINAMICS → Overview → Computing time load, memory requirement and assignment of the HW sampling times → Number of possible different hardware sampling times (Page 169).

Note

As regards offline configuration using STARTER commissioning software, values 0 - 256 can be entered in p21000[0..9], even if this violates the limits stated above for the hardware sampling times from 1 ms ... <r21003. This will only be detected once the Control Unit has been downloaded, and will result in the use of a replacement value and the F51004 fault. If the set value is too small, 1 ms will be set as substitute value; if it is too large, the next largest SW sampling time will be set.

   – On the drive objects of the CU, TB30, TM15 DIDO, TM 31 and TM41, the sampling time for additional functions p0115[0] = 4 ms is preset. If you want to configure a DCC execution group with a smaller sampling time on these drive objects, you should first set the sampling time for the additional functions p0115[0] on this drive object to the value of the smallest desired sampling time. To do this, p0009 = 3 must first be set. Only then is it possible to change p0115[0]. For the new value for p0115[0] to take effect, p0009 = 0 must be reset.

● Free execution groups whose sampling times are created in the software:
   The software sampling times are generated as an integer multiple of the basic value for software sampling times (read out in parameter r21003).
   The possible set values for the software sampling times (1 * r21003 - 96 * r21003) can be taken from the parameter description for p21000 (see Function Manual, *Description of the standard DCC blocks*, in the appendix in the *Parameters* section).

⚠ **WARNING**

The START method of the block is called for an online change to the sampling time (execution group). Initializations in the START method can result in jumps in the signal path.

4.1.3.5 Execution sequence order, creating new execution groups

When a CFC is created, a new execution group with the same name as the chart will be created automatically. All blocks that are inserted into the chart will be automatically assigned in the insertion sequence of this execution group.

The computing sequence of the blocks within the execution group is specified by the configuration. If no other execution sequence is specified by the user, the execution sequence corresponds to the insertion sequence of the blocks within an execution group.

If the execution sequence with this execution group is to be changed, new execution groups added or the assignment of blocks to the execution groups changed, these properties can be changed in the execution editor. To do this, go offline in the STARTER and then open the chart. In the DCC editor, the **execution sequence** button can be used to activate and deactivate the execution editor.

The center column of the execution editor contains the chart with its chart name. The execution group(s) are arranged hierarchically below. If you want to create additional execution groups, select the **Insert execution group…** item from the context menu of the chart (right-click the chart). The individual blocks can be moved between the execution groups or within an execution group using drag-and-drop. The details given in the Pos column (e.g. 1 / 2) correspond to the name in the bottom line in the right-hand part of the block header with a colored background. The 1st digit stands for the execution group (between 1… 10) and the 2nd digit for the sequence within the execution group. The execution sequence can be changed by dragging & dropping into the center column.
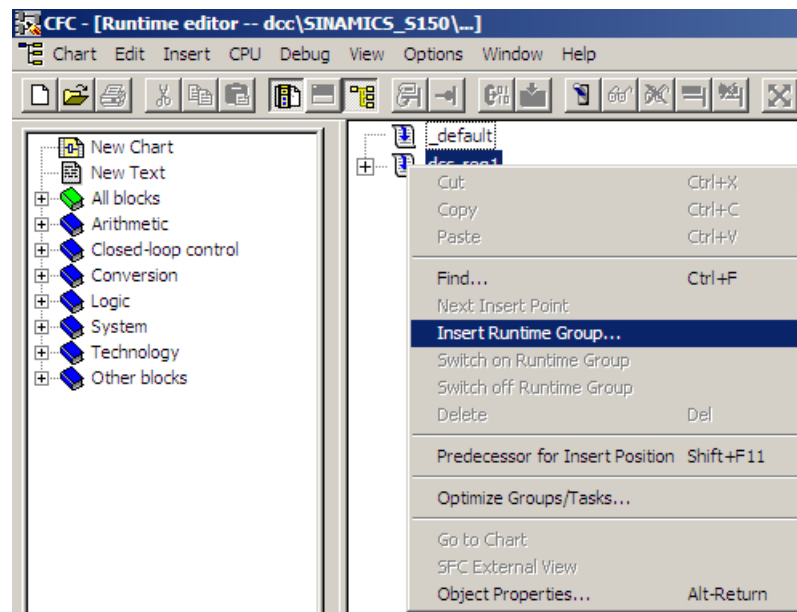
Figure 4-4   Context menu, **Insert execution group...**

Note

1. The assignment of identical names for DCCs in different drive objects is not permitted.
2. The assignment of identical names for execution groups in different drive objects is not permitted.

If the same fixed or free execution group of the drive object is assigned to several execution groups in the DCC editor, the execution groups of the DCC editor will be calculated in the sequence from top (calculated first) to bottom (calculated last) as shown in the "Set Execution Groups (Sampling Times)" window.

4.1.3.6 Creating customer-specific parameters ("declare")

The properties of the input and output connections of blocks can be edited in the DCC editor in the Properties window of the block connection. Input connections may not be interconnected in the default setting of the DCC editor. The Properties window of the connection is opened by double-clicking the connection or by selecting **Object properties** in the context menu of the connection.
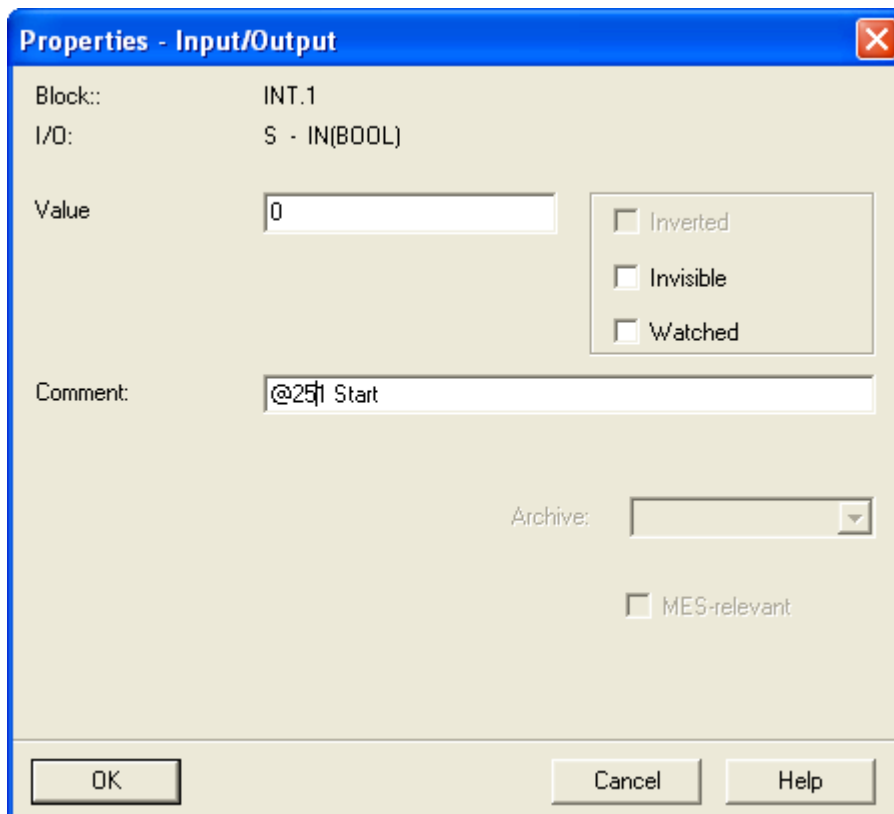
Figure 4-5   **Properties - Connection** window of the connection S (= Set) of the block INT (= INTegrator)

In SINAMICS, input and output connections of blocks can be published as parameters. "Publish" means that users create the parameters themselves (parameter numbers and parameter text). This is the prerequisite in order to interconnect the block connections using BICO parameters to the basic system in order to assign the values to the input connections using adjustable parameters and to monitor the values of the output connections using parameters. The parameter number 251 in the above figure and in the following figure is arbitrary.
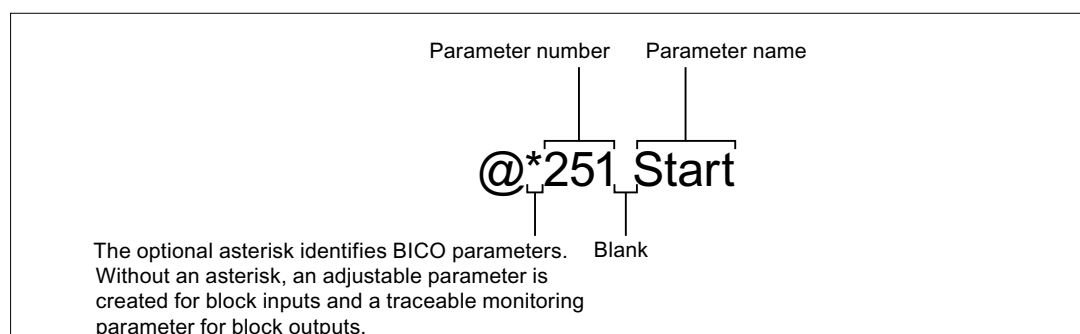
Figure 4-6   Structure of the block connection comment

The parameter name, which follows the parameter number separated by a space, is displayed in the STARTER expert list.

The data type of the published input/output is taken over by the block connection.

For the "publishing" of block inputs, a distinction must be made between adjustable parameters (without asterisk) and BICO parameters (with asterisk).

For the "publishing" of block outputs, the following distinction must be made:

- Without asterisk: pure monitoring parameter that cannot be interconnected and that can be recorded with the trace function.

- With asterisk: as for the case without asterisk, but additional interconnection possible.

Block inputs to be published may not be already interconnected in the default setting of the DCC editor. If necessary, an existing interconnection must first be deleted. To do this, click the interconnection in the chart to mark it and select **Delete interconnection(s)** in the context menu. In this DCC editor setting, the comment shown for a block input interconnected with a different block is always that of the connected block output.

If you want to remove this constraint, select **Options > Settings > Display ...** in the DCC editor and deactivate the **Interconnection comment** checkbox under **Parameters** and close the window with OK. Already interconnected block inputs can now be published without having to delete the interconnection first.

A parameter number may only be used just once in a chart. The DCC editor does not check when the parameter number is entered in the Properties window of the connection whether this parameter number is already used in the chart. The multiple use of the parameter numbers will be indicated as error only when the chart is compiled.

The parameter numbers of the charts are represented on the number range from p21500 to p25999 of the expert list of the STARTER. With each DCC chart, it is possible to specify where

the parameter range of the chart is to begin. This can be performed in the **Parameters** tab in the Properties window of the chart in STARTER, using the parameter number base.
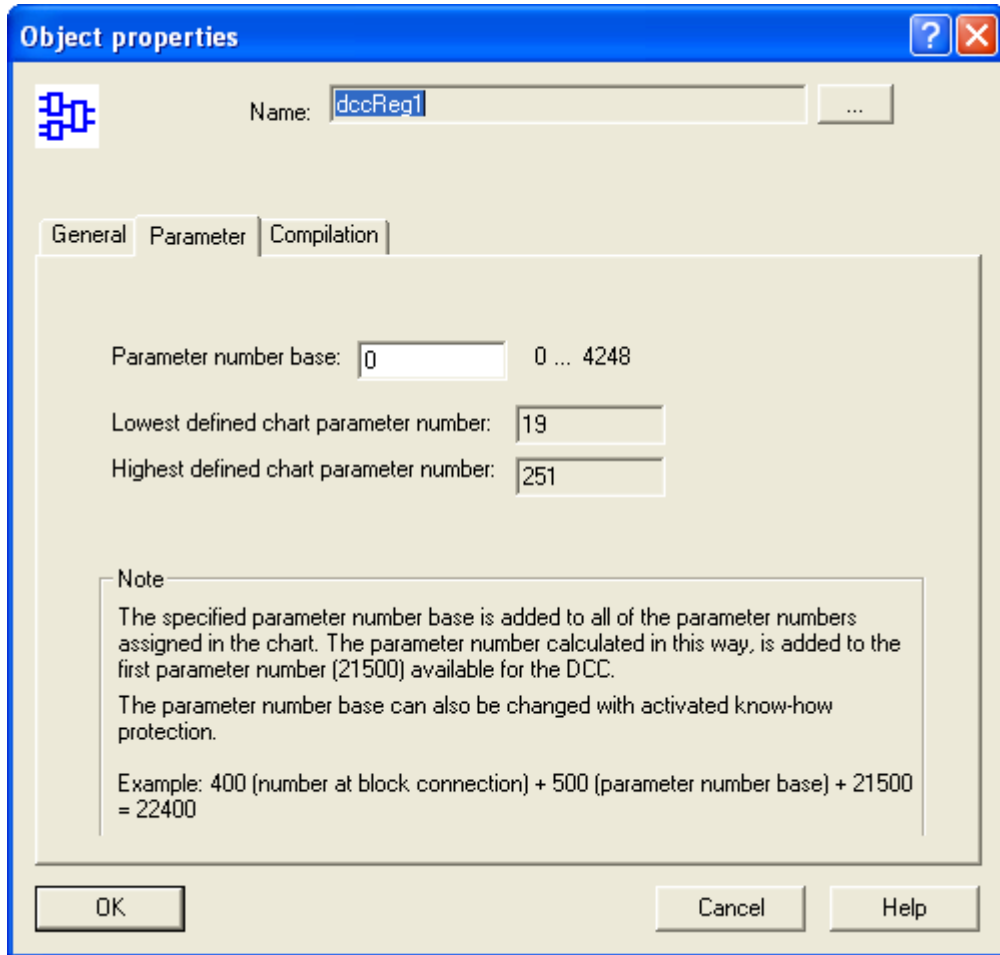
Figure 4-7   Properties window of the dccReg1 chart for setting the parameter number base

The parameter number in STARTER has the following form:

- Parameter number in the chart + base parameter number + 21500;
where 21500 is the first parameter number available for DCC. The parameter number base must always be >= 0. The parameter number base can be changed.
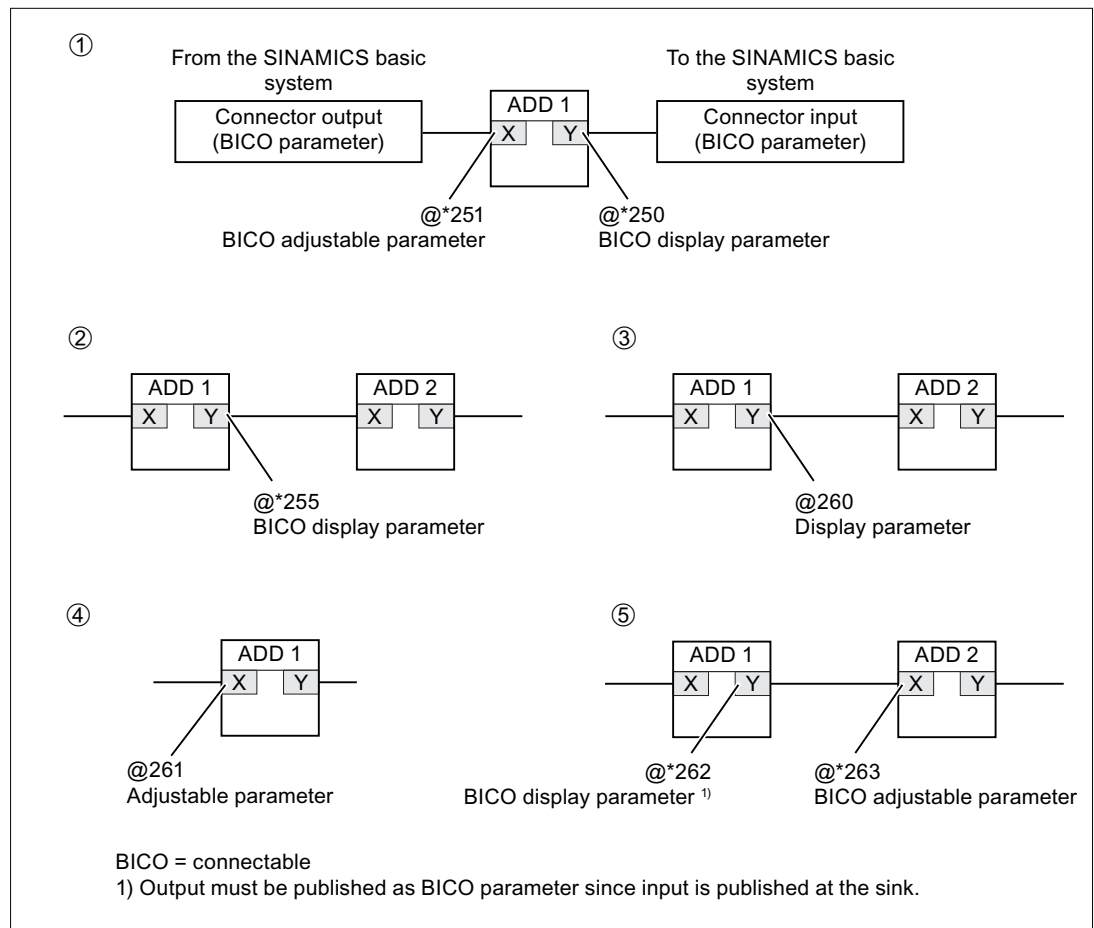
Figure 4-8   Examples of parameter definitions

Example 1: For an interconnection of the block connection with the SINAMICS basic system, the connection must be published as BICO parameter.

Example 2: Each block output connection can be published as BICO parameter. The signal on this output can be recorded with the trace.

Example 3: Each block output connection can be published as (not interconnectable with other BICO parameters) display parameter. The signal on this output can be recorded with the trace. The block input can be interconnected to every other non-published block input.

Example 4: The block input connection X of the ADD 1 block is published as adjustable parameter.

Example 5: Because the block input connection X of the ADD 2 is published as BICO parameter, the block output connection interconnected with it must also be published as BICO parameter. It is, however, possible in the DCC editor to interconnect ADD 1 Y with ADD 2 X when only the block input ADD 2 X is published. This results in an error message when the

chart is compiled. The signal value of a block input published as BICO parameter cannot be recorded with the trace.

Note

No help function can be created for @parameters.

Display of the parameters

If the parameters configured on the block connections are to be displayed in the DCC editor, then select **Options > Settings > Display ...** and select **Comment** under **Connections**. The first eight characters of the comment are then displayed at the respective block connections in the block symbols.

4.1.3.7   Interconnection with SINAMICS parameters

You can interconnect with SINAMICS BICO parameters in the DCC editor.

Procedure

Interconnection can be performed as follows:

1. Creating customer-specific parameters ("Publishing", @ parameters); see Creating customer-specific parameters (Page 156).

Note

In the DCC editor, block connections can initially be connected with the drive's BICO parameters without the need to publish them as parameters. However, this will cause errors when compiling the chart, since only block connections that have been published as @* parameters can be connected to the drive's BICO parameters.

---

2. Select the block connection to be interconnected.

3. Right-click and select **Interconnection with operand...** from the context menu. The **DCC Signal Selection** window appears.
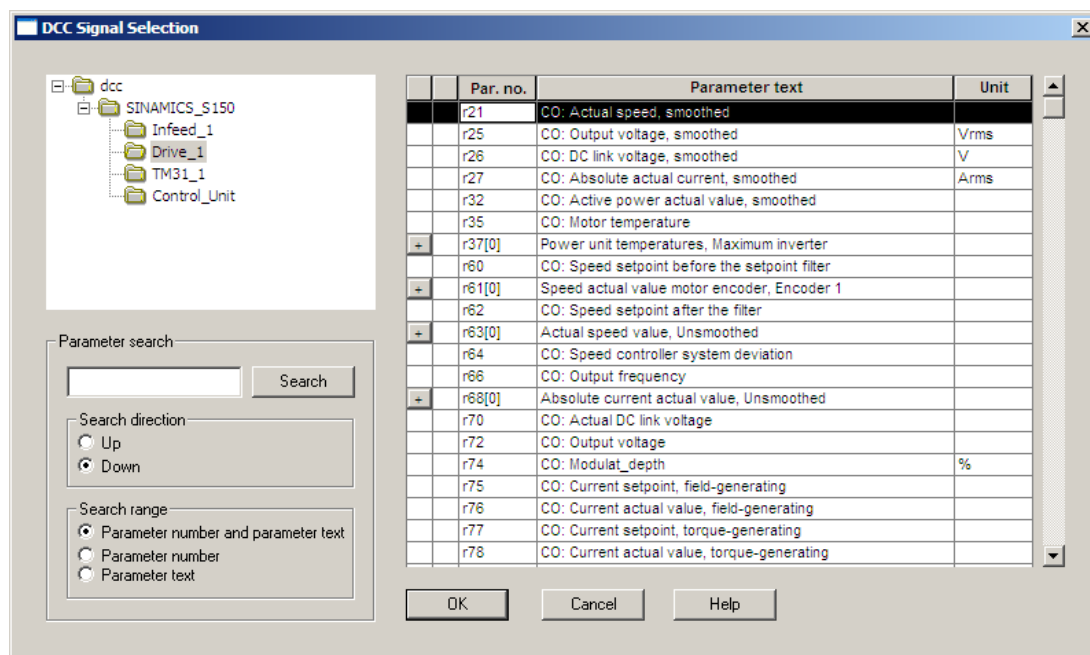
Figure 4-9   Signal selection

4. Select the parameter to be interconnected.

5. Click **OK** to close the window.

From this point on, the selected block connection is interconnected with the SINAMICS parameter and the parameter is displayed in the sheet bar of the DCC editor.

Note

BICO parameters defined in other DCC charts will not be displayed in the signal selection list.

Connections to BICO parameters in DCC charts on other DOs can be established graphically in the DCC editor using two charts open next to one another.

Note

With the BICO technology of the SINAMICS drive family you can interconnect BICO parameters with different data types, e.g. a BICO parameter of the FLOAT32 data type with a BICO parameter of the INT32 data type. In this case, the value range of the FLOAT32 data type exceeds the value range of the INT32 data type.

The SINAMICS basic system automatically ensures that in such cases the source parameters do not exceed the value ranges of the sinks.

You must explicitly ensure this restriction in the DCC chart for published BICO parameters, e.g. through an LIM block.

4.1.4 Computing time load, memory requirement and assignment of the HW sampling times

The following statements govern the use of DCC-SINAMICS on the SINAMICS S120, S150, G130, G150 and (on the SINAMICS_Integrated in) SIMOTION D4xx devices. Details about other devices are contained in the associated device documentation.

4.1.4.1 Computing time load for the SINAMICS V2.5 and V2.6 software versions

The processing of blocks requires computing time. This means with the computing of DCC execution groups, it is no longer possible to calculate the maximum quantity structure for drive objects of a CU3x0, D4xx and CX32.

The resulting computing time load is dependent on the following:

- Number of calculated execution groups
- Sampling time of execution groups
- Number of calculated blocks
- Calculated block types
- For some blocks, it depends on the parameterization (e.g. the setting of enable connections)

Parameter r9976 (system load; can be found on the CU's drive object) is available in the system as an online tool. The average value for the computing time load in r9976[1] should **always** be below 85.0%. Likewise, the maximum value for the computing time load in r9976[5] should **always** be below 85.0%. Exceeding the computing time load limit of 90% triggers warning A51003. The warning is reset if the computing time load falls below 88%.

The computing time load should not fall below the limit of 85.0% so that the response time of the STARTER/SCOUT engineering tool is not too greatly restricted.

If the computing time load increases to the point that it is no longer possible for all blocks and system functions assigned to a sampling time to be completed within this sampling time, the drive unit switches off and outputs fault F01205: "CU: Time-slice overflow".

Offline mode enables the SIZER engineering tool to make an approximate statement regarding whether a configuration can be calculated on a CU3x0 or D4xx. The additional computing time load is **not** taken into account by activating the DCCs.

Note

The computing time load reaches its maximum value when the following conditions are met:
- All necessary function modules are activated.
- All drives and infeeds are in the operational status.
- Closed-loop control in the final configuration is approved on all drives and infeeds.
- Configured isochronous data transfer is used.

The values displayed in r9976 are thoroughly smoothed internally. A change in the computing time load is therefore only displayed in full in r9976 after 2 - 3 minutes.

If, during OFFLINE configuration, it is not certain whether the limit value for the computing time load of 85% (r9976[1] and r9976[5]) will be adhered to, you can initially call the execution groups being used in a very large sampling time (e.g. 96 * r21003). After checking the actual system load in r9976, you can then gradually select execution groups with shorter sampling times right through to the execution group that is intended for the application in question.

During OFFLINE configuration, the following should be carried out:

● Activate all necessary function modules on the drive objects.

● Assign all necessary blocks to their execution groups.

● Establish all connections, both between the blocks and to the drive objects (as long as this still makes sense even if an increased sampling time is being used).

The following applies as a rough configuration aid to SERVO drives:

If one A-INF infeed is configured on a SINAMICS CU320 with the sampling times set in the factory (p0112 = 2 == LOW → p115[0] = 250 μs) and three SERVO drive axes are configured with the speed and current controller sampling times set in the factory (p0112 = 3 == STANDARD → p0115[1] = p0115[0] =125 μs) and the function modules and without isochronous PROFIBUS/PROFINET communication set in the factory, one DCC chart with 50 blocks can be calculated on each of the three drive objects in an execution group with a sampling time of 2 ms.

The following (functional single) blocks have been used in a chart in this case:

AND, OR, XOR, NOT, BF, CTR, MUX8, NAND, NCM, SH_DW, ADD, SUB, MUL, DIV, AVA, MAS, MIS, PLI20, DEL, DEZ, SUB_D, SUB_I, MUL_D, MUL_I, DIV_D, DIV_I, AVA_D, SII, STM, DLB, DX8, DX8D, MFP, PCL, PDE, PDF, PST, RSR, DFR, BSW, NSW, LIM, PT1, INT, DIF, LVM, LIM_D, PC, PIC, RGE

In a sampling time of 1 ms, only around half the blocks can be calculated. Around twice as many can be calculated in a 4 ms sampling time.

A SERVO axis with standard sampling times omitted from the maximum configuration thus permits the use of approximately 50 blocks (of the above-mentioned types) for a sampling period of 2 ms.
Approximately 50 blocks (of the above-mentioned types) in an execution group increase the computing time load of the CU by approximately 11%.

The following applies as a rough configuration aid to VECTOR drives:

If one A-INF infeed is configured on a SINAMICS CU320 with the sampling times set in the factory (p0112 = 2 == LOW → p115[0] = 250 μs), two VECTOR drive axes are configured with the speed controller sampling times p0115[1] = 2000 μs and the current controller sampling times p0115[0] = 500 μs (p0115[2] = p0115[3] = p0115[4] = 2000 μs) and the function modules and without isochronous PROFIBUS/PROFINET communication set in the factory, one DCC chart with approx. 70 blocks (type selection, see above for SERVO) can be calculated on two drive objects in an execution group with a sampling time of 2 ms.
This means approx. 70 blocks (type selection, see above for SERVO) increase the computing time load by approx. 15%.

In a sampling time of 1 ms, only around half the blocks can be calculated. Around twice as many can be calculated in a 4 ms sampling time.

The following generally applies:

You can of course use any blocks you want in your project. Using other block types may produce other results in terms of the number of blocks that can be calculated and the computing time load. Blocks with a very extensive functionality of course require a greater computing time. Note that the computing time needed for some blocks also depends on the values configured at the connections, e.g. whether an enable connection is set.

When measuring the computing time load, all connections should therefore be set to their final configuration values wherever possible.

When commissioning a configuration, the current computing time load must always be checked online on the drive unit for compliance with the limits stated at the start for r9976[1] and r9976[5].

4.1.4.2 Computing time load as of software version V4.3

With software version SINAMICS V4.3, DCC is available for the new higher performance CU320-2 DP module. It is also available for the new modules CU310-2 DP, CU310-2 PN, and CU320-2 PN in the case of SINAMICS V4.4, and for SIMOTION D4x5-2. As of software V4.3, after a download (DL) or a parameter change (e.g. of the sampling time of a execution group), the control unit (CU) uses the configuration data to determine the expected computing time utilization (including the load due to DCCs) and indicates this for the complete system in parameter r9976 (the level of system utilization that can be found on the drive object of the CU). If the calculated average computing time utilization for the complete system r9976[1] or the maximum utilization in a sampling time r9976[5] (including interruptions due to shorter sampling times) exceeds the maximum value of 100.00%, this causes the setting of fault F1054 (CU: System limit exceeded). The fault F1054 causes all infeeds and drive axes calculated on the CU to be switched off. The utilization is calculated on the CU, i.e. the utilization values are displayed in the STARTER/SCOUT only in online mode.

The resulting computing time utilization caused by DCC depends on the following:

- Number of calculated execution groups

- Sampling time of the execution groups

- Number of calculated blocks

- Calculated block types

- Use of the execution groups "Receive AFTER IF1 PROFIdrive PZD ", "Send BEFORE IF1 PROFIdrive PZD", "Receive AFTER IF2 PROFIdrive PZD", and "Send BEFORE IF2 PROFIdrive PZD" of the bus configuration (isochronous/ not isochronous; see chapter Fixed execution groups (Page 141))

The proportional computing time utilization caused by DCC is displayed on the drive objects on which DCCs are configured in the parameter r21005[0…9] for the execution groups 1 to 10 (as of SINAMICS V4.3). Note that the average computing time utilization for an execution group k can be calculated only when this is registered for cyclical processing (p21000[k-1] ≠ 0, in the STARTER/SCOUT in the context menu of the chart → Set sampling times).

In contrast to software versions V2.5 and V2.6, a parameter change (in online mode of the STARTER) that affects the computing time utilization (e.g. change of the sampling time of a DCC execution group) causes an immediate recalculation of r9976 (and r21005) by the drive device. For parameters that may only be changed in the device states C1 (commissioning 1) or C2 , i.e. may only be changed in STARTER/SCOUT offline mode (e.g. p0115), the new value of the computing time utilization in r9976 will be displayed only after the project download and the subsequent CU ramp-up.

As of V4.3, the permitted maximum value of the computing time utilization r9976 increases practically to 99.99%. (For values > 100.00%, fault F1054 will be set; this simultaneously causes an OFF2 on all drive objects of the CU.)

As rough configuration aid, the following is true for **SERVO** drives with the current controller sampling time p0115[0] = 125 µs and the speed controller sampling time p0115[1] = 125 µs:

1. If on a SINAMICS CU320-2 one A-INF infeed with the sampling times set in the factory (p0112 = 2 == LOW → p115[0] = 250 µs), **six SERVO drive axes** with the speed and current controller sampling times set in the factory (p0112 = 3 == STANDARD → p0115[1] = p0115[0] =125 µs), the function modules set in the factory setting, one TB30 [2] and with isochronous PROFIBUS/PROFINET communication with T_{DP} >= 2 ms are configured, one (1) DCC with approximately 50 blocks can be calculated in an execution group with a sampling time of 2 ms.

2. If on a SINAMICS CU320-2 one A-INF infeed with the sampling times set in the factory (p0112 = 2 == LOW → p115[0] = 250 µs), **5 SERVO drive axes** with the speed and current controller sampling times set in the factory (p0112 = 3 == STANDARD → p0115[1] = p0115[0] =125 µs), the function modules set in the factory setting, 1 TB30 [2], 3 TM31 [2] and isochronous PROFIBUS/PROFINET communication with T^{DP} >= 2 ms are configured, one (1) DCC chart with approximately 75 blocks can be calculated [1] in an execution group with a sampling time of 2 ms.

3. For each additional omitted SERVO axis, approximately 75 additional blocks [1] can be calculated in the sampling time of 2 ms.

As rough configuration aid, the following is true for **VECTOR** drives with the current controller sampling time p0115[0] = 500 µs and the speed controller sampling time p0115[1] = 2000 µs (p0115[3] = 2000 µs, p0115[4]=2000 µs):

For more than three VECTOR axes on a CU320-2, the current controller sampling time will be increased automatically to the value p0115[0] = 500 µs. This automatically increases the speed controller sampling time to 2 ms.

1. If, on a SINAMICS CU320-2, one A-INF infeed with the sampling times set in the factory (p0112 = 2 == LOW → p115[0] = 250 µs), **six VECTOR drive axes** with the current controller sampling times p0115[0] = 500 µs, the speed controller sampling time p0115[1] = 2000 µs, the functions and function modules set with the factory setting, one TB30 [2] and with isochronous PROFIBUS/PROFINET communication with $T_{\_DP}$ >= 2 ms are configured, one (1) DCC with approximately 50 blocks can also be calculated in an execution group with the sampling time 2 ms.

2. If, on a SINAMICS CU320-2, one A-INF infeed with the sampling times set in the factory (p0112 = 2 == LOW → p115[0] = 250 µs), **five VECTOR drive axes** with the current controller sampling times p0115[0] = 500 µs, the speed controller sampling time p0115[1] = 2000 µs, the functions and function modules set with the factory setting, with isochronous PROFIBUS/PROFINET communication with $T_{\_DP}$ >= 2 ms, one TB30 [2] and three TM31 [2] are configured, one (1) DCC with approximately 75 blocks can also be calculated in an execution group with the sampling time 2 ms.

3. For each additional omitted VECTOR axis (with p0115[0] = 500 µs and p0115[1] = 2000 µs), approximately 75 additional blocks [1] can be calculated in the sampling time of 2 ms.

As rough configuration aid, the following is true for **VECTOR** drives with the current controller sampling time p0115[0] = 250 µs and the speed controller sampling time p0115[1] = 1000 µs (p0115[3] = 1000 µs, p0115[4] = 1000 µs):

For up to three VECTOR axes on a CU320-2, the current controller sampling time and the speed controller sampling time have the values p0115[0] = 250 µs and 1 ms, respectively, in the factory setting.

1. If, on a SINAMICS CU320-2, one A-INF infeed with the sampling times set in the factory (p0112 = 2 == LOW → p115[0] = 250 µs), **three VECTOR drive axes** with the current controller sampling times p0115[0] = 250 µs, the speed controller sampling time p0115[1] = 1000 µs, the functions and function modules set with the factory setting, one TB30 [2] and with isochronous PROFIBUS/PROFINET communication with $T_{\_DP}$ >= 2 ms are configured, one (1) DCC with approximately 50 blocks can also be calculated in an execution group with the sampling time 2 ms.

2. If, on a SINAMICS CU320-2, one A-INF infeed with the sampling times set in the factory (p0112 = 2 == LOW → p115[0] = 250 µs), **two VECTOR drive axes** with the current controller sampling times p0115[0] = 250 µs, the speed controller sampling time p0115[1] = 1000 µs, the functions and function modules set with the factory setting, with isochronous PROFIBUS/PROFINET communication with $T_{\_DP}$ >= 2 ms, one TB30 [2] and three TM31 [2] are configured, one (1) DCC with approximately 150 blocks can also be calculated in an execution group with the sampling time 2 ms.

3. For each additional omitted VECTOR axis (with p0115[0] = 250 µs and p0115[1] = 1000 µs), approximately 150 additional blocks [1] can be calculated in the sampling time of 2 ms.

The following generally applies:

You can of course use any blocks you want in your project. The use of other block types may produce different results in terms of the number of blocks that can be calculated and the computing time utilization. Blocks with a very extensive functionality of course require a greater computing time.

[1] The chart with 75 blocks consists of an execution group with the following blocks in the listed sequence:

AND, OR, XOR, NOT, BF, CTR, MUX8, NAND, NCM, SH_DW, ADD, SUB, MUL, DIV, AVA, MAS, MIS, PLI20, DEL, DEZ, SUB_D, SUB_I, MUL_D, MUL_I, DIV_D, DIV_I, AVA_D, SII, STM, DLB, DX8, DX8_D, MFP, PCL, PDE, PDF, PST, RSR, DFR, BSW, NSW, LIM, PT1, INT, DIF, LVM, LIM_D, PC, PIC, RGE, AND, OR, XOR, NOT, BF, CTR, MUX8, NAND, NCM, SH_DW, ADD, SUB, MUL, DIV,AVA, MAS, MIS, PLI20, DEL, DEZ, SUB_D, SUB_I, MUL_D, MUL_I, DIV_D.

The chart with 150 blocks consists of an execution group that contains the previously listed 75 blocks twice.

[2] All sampling times (p4099[]) of the TB30 and the TM31 have the value 4 ms (factory setting).

4.1.4.3   Memory requirement

The blocks and @ parameters in the DCC charts require memory on the drive unit. To use DCC-SINAMICS, for SINAMICS G130, G150, S120, S120 Chassis, S120CM, S150 and SIMOTION D4x5, at least one drive axis must be excluded in comparison to the maximum possible configurations (1 ALM + 6 servo axes + TB30 or 1 ALM + 4 vector axes + TB30).

Maximum quantity structures with a topology of 5 servo or 3 vector axes with 1 ALM and 1 TB30			
Drive unit		CU320[2]	SINAMICS Integrated on SIMOTION D4x5
SINAMICS 2.6.1	Blocks[1]	350	-
	@Parameters[1]	350	-
SINAMICS 2.6.2	Blocks[1]	500	200
	@Parameters[1]	500	200

2) SINAMICS G130, G150, S120, S120 Chassis, S120CM, S150

Note

The specified maximum numbers for blocks and @parameters always apply for the entire drive unit and should be regarded as guide values. Only the factory-set function modules are active. The individual blocks and @parameters may be arbitrarily distributed across several charts. Conserving @parameters has little effect on the quantity structure of the blocks; the specified maximum numbers for the blocks should therefore not be exceeded.

Savings in further drive axes can increase the maximum limits by 50 blocks and 50 @parameters for each axis saved.

Maximum limits when just one axis is used (including 1 ALM + TB30)					
Drive unit		CU320 and CU310[2] with 1 servo axis	CU320 and CU310[2] with 1 vector axis	SINAMICS Integrated[3] with one servo axis	SINAMICS Integrated[3] with one vector axis
SINAMICS 2.6.1	Blocks[1]	550	450	-	-
	@Parameters[1]	550	450	-	-

Maximum limits when just one axis is used (including 1 ALM + TB30)					
Drive unit		CU320 and CU310[2] with 1 servo axis	CU320 and CU310[2] with 1 vector axis	SINAMICS Integrated[3] with one servo axis	SINAMICS Integrated[3] with one vector axis
SINAMICS 2.6.2	Blocks[1]	700	600	400	300
	@Parameters[1]	700	600	400	300

2) SINAMICS G130, G150, S120, S120 Chassis, S120CM, S150

3) S120i on D4x5

When expanding the topology on a drive unit with TMxx or DMC20 modules, one drive unit must be omitted for each additional TMxx module.

Please note when connecting CX32 modules to a D4x5 that memory and CPU time are also allocated for each CX32 on the D4x5. When using the DCC, another servo or vector axis must be omitted for every 2 CX32s (i.e. the axis must be omitted for the first CX32).

Example for SINAMICS 2.6.2:

D4x5 with 1 ALM + 4 servo axes + 2 CX32 + 1TB30 à
 Permits DCC with 200 DCBs + 200 @parameters.

D4x5 with 1 ALM + 2 vector axes + 2 CX32 + 1TB30 à
 Permits DCC with 200 DCBs + 200 @parameters.

When using DCC on the CX32 the same quantity structure applies as on the CU320.

Note

The final limits are determined by the total memory available on the drive unit and the computing time load. If the above-mentioned recommended maximum limits are exceeded, this can result in errors during upload or download (e.g. fault F1105 CU: insufficient memory) and the drive can no longer be switched on. If it is not possible to perform a new download with a suitably adjusted project, a power OFF/ON must be performed on the affected drive unit.

By activating additional function modules (e.g. basic positioner EPOS), additional memory is allocated and the specified maximum limits are decreased.

---

[1]) The charts for the above-specified quantity structures are formed from n x (chart1 + chart2): Chart1 contains 50 blocks (of the following types: AND, OR, XOR, NOT, BF, CTR, MUX8, NAND, NCM, SH_DW, ADD, SUB, MUL, DIV, AVA, MAS, MIS, PLI20, DEL, DEZ, SUB_D,SUB_I, MUL_D, MUL_I, DIV_D, DIV_I, AVA_D, SII, STM, DLB, DX8, DX8D, MFP, PCL, PDE, PDF, PST, RSR, DFR, BSW, NSW,LIM, PT1, INT, DIF, LVM, LIM_D, PC, PIC, RGE) with 50 @parameters and approx. 90 connections (from block connector to block connector).

Chart2 contains 50 blocks (of block types AND, OR, ADD, MUL, DIV, B_DW, B_W, BY_W, D_I, D_R, D_UI, D_US, DW_B, DW_R, DW_W, I_D, I_R, I_UD, I_US, N2_R,N4_R, R_D, R_DW, R_I, R_N2, R_N4, R_UD, R_UI, R_US, UD_I, UD_R, UI_R, US_D, US_I, US_R, W_B, W_BY, W_DW, WBG,DCA, INCO, OCA, TTCU, ADD, ADD_D, ADD_I, ADD_M, AVA, AVA_D, RSS), 50 @parameters and approx. 90 connections (from block connector to block connector)

For example, 350 blocks and 350 @parameters are compiled from 4 * chart1
+ 3 * chart2.

4.1.4.4   Memory requirement as of software version SINAMICS V4.3

The blocks and @ parameters in the DCC charts require memory on the drive unit. A maximum of 1,500 blocks and 1,500 @ parameters may be configured in DCC SINAMICS with the modules CU320-2 DP, CU320-2 PN (V4.4 or higher), CU310-2 DP (V.4.4 or higher), and CU310-2 PN (V.4.4 or higher) on SINAMICS S120, S150, G130, and G150.

For SINAMICS_Integrated V4.4 or higher (V4.2 or higher with SIMOTION), a maximum of 1,500 blocks and 1,500 @ parameters may be configured on the D4x5-2 modules.

4.1.4.5   Number of possible different hardware sampling times

The sampling times for the execution groups can be selected in p21000[x] as a multiple of r21002 (basic sampling time of hardware time slices), a multiple of r21003 (basic sampling time of software time slices), or on the basis of the sampling time of a basic SINAMICS system function (e.g. when p21000[x] = 9003 == "before setpoint channel" from the sampling time of the setpoint channel p0115[3]).

Only sampling times can be set as hardware sampling times, for which the following applies:

1 ms <= T_sampling <= r21003 - r21002 in p21000[x]

Hardware sampling times, assignment, and number

The assignment of the available hardware sampling times is displayed in r21008[0...24] as follows (STARTER/SCOUT: in online mode only):

- Value = 0.0 --> sampling time not assigned

- Value != 0.0 (not equal to 0.0) --> sampling time in ms

- Value = 9.9999e + 006 --> sampling time not supported

During configuration, note that the number of different hardware sampling times (1 ms = cycle duration T_sampling <r21003 - r21002) used by the basic SINAMICS system, active function modules (see r108) and Drive Control Chart is restricted as follows:

- CU310, CU320, D4xx --> no. of hardware sampling times = 13

- CU320-2 DP with SINAMICS V4.3 and higher --> no. of hardware sampling times = 25

- CU310-2 DP, CU310-2 PN, and CU320-2 PN with SINAMICS V4.4 and higher --> no. of hardware sampling times = 25

- SINAMICS_Integrated on the D4x5-2 with SINAMICS V4.4 (contained in SIMOTION V4.2) and higher --> no. of hardware sampling times = 25

- CUD (SINAMICS DC MASTER) --> number of hardware sampling times = 11

Hardware sampling times, usage

A sampling time can be used simultaneously by more than one execution group of DCC, function modules and the basic SINAMICS system.

For this reason, the execution groups should ideally be registered to existing sampling times or, if it makes more sense in relation to the function, the execution group "Before setpoint channel" should be used.

For internal reasons, the drive unit always requires at least one (or more, depending on how the basic sampling times p0115[0] of the drive objects are parameterized) free hardware sampling time, which is why the current **Number of free hardware sampling times** can be read in **r7903** (as of SINAMICS V2.6).

DCC and FBLOCKS together must not use more than 5 different hardware sampling times on one drive unit. (This only refers to sampling times that deviate from the hardware sampling times already present in the basic system.)

Note

Note that a long-term trace registers a sampling time of 2 ms and the trace registers sampling times in accordance with the selected trace cycle clock. If these sampling times have not already been registered by the basic SINAMICS system, "Free blocks" (FBLOCKS), or Drive Control Chart (DCC), these functions require additional free hardware sampling times.

The registered hardware sampling times can be read (if the DCC is activated) in r21008[0...12].

The current number of free hardware sampling times is displayed in r7903 (as of SINAMICS V2.6).

Project download, error message, and procedure

If too many different hardware sampling times are configured offline, an error message is not output until the project is downloaded.

In this case, proceed as follows:

1. With the project in offline mode, set all the free execution groups to which hardware sampling times are assigned to software sampling times.

   – Hardware sampling times (DCC: p21000 < 256)

   – Software sampling times (DCC: p21000 >= 1001)
     The assignment of fixed execution groups (DCC: p21000 >= 2000) does not need to be changed because the fixed execution groups use the same sampling time as the assigned basic SINAMICS system function.

2. Download the project again.

3. Once the project has been downloaded and the Control Unit has ramped up, check:

    – r7903: Number of hardware sampling times still available

    – r21008: Hardware sampling times already registered by the basic SINAMICS system.

4. Adjust the execution group parameters accordingly.

Note

The number of different hardware sampling times possible on a Control Unit is restricted. For this reason, software sampling times (multiple of r21003) or, if necessary, the fixed execution group (p21000[0...9] ≥ 2000) should ideally be used.

4.2 Working with DCC SINAMICS

4.2.1 Preliminary remarks on configuration

Preliminary remarks

The following is a brief explanation of what you will be configuring in this chart.

Configuration example

The requirement for this example is that there must be a STARTER or SCOUT version (≥ V4.1.2) suitable for your SINAMICS SW version (≥ V2.6) installed on your computer. The SINAMICS Support Package (SSP) V2.6 must also be installed. Your computer must also have a CFC license to use the DCC editor. You can install this license from the USB stick provided with the help of the Step7 Automation License Manager. A CU310 or CU320 is required with which STARTER/SCOUT can connect online, e.g. via PROFIBUS.

The configuration example deals with a straightforward oscillating circuit that creates sinusoidal oscillation at its output.

It will only take you a few minutes to create the chart; then you can execute it in test mode as a demonstration.

The following blocks are used:

- Two integrators (**INT**)
- One inverter (**SII**)

As indicated by the differential equation $f''(x) = -f(x)$, the oscillating circuit is comprised of two integrator blocks that are linked by negation.

The frequency of the oscillating circuit is determined by the integrating time constant at the integrators.

The oscillation amplitude is specified by the initial value at the integrator output.

Configuration example structure

The configuration example is divided into the following steps:

1. Creating a new project.
2. Inserting the DCC in the project.
3. Inserting blocks in a DCC.
4. Interconnecting blocks.
5. Parameterizing block connections in the chart.
6. Publishing block connections as parameters.
7. Setting execution sequence within an execution group
8. Compiling the DCC in the DCC editor.

9. Setting sampling time for an execution group

10. Loading the DCC technology option on to the drive unit's CF card.

11. Downloading the compiled DCC to the drive.

12. Displaying values at block connections online.

13. Recording signals from the DCC with the trace.

14. Archiving the project.

15. Creating documentation.

4.2.2 Creating a new project

- Create a new project in the SCOUT or STARTER engineering system, e.g. **dcc_ex**, see Creating a project (Page 23).

- Create a new device. To do this, insert a "SINAMICS S120 CUxxx" in the STARTER or SCOUT by double-clicking the "Insert single drive unit" command. Select "xxx" to suit your existing hardware (CU320, CU310DP, CU310PN).

- You can now insert a chart.

4.2.3   Inserting a DCC

- Open up the tree structure in the project navigator to the control unit.

- Call **Insert DCC** below the control unit (double-click).
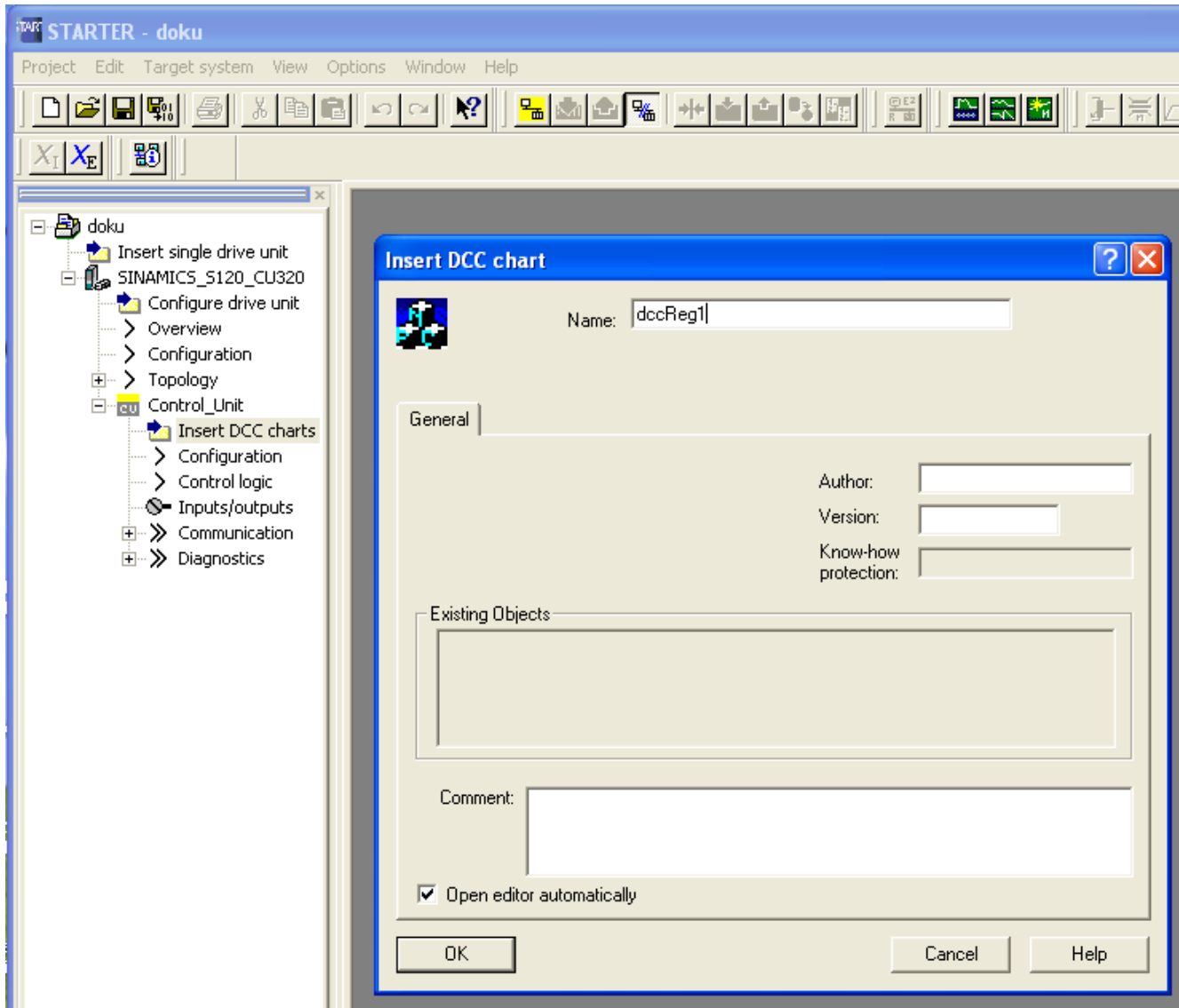
Figure 4-10   Inserting a DCC SINAMICS chart

- The **Insert DCC** window opens and you can enter a new name (of no more than 22 characters) for the chart (in our example *dccReg1*) and a comment. Please note that all characters after an underscore in the chart name can only be digits.

- Click OK to close the window. The DCC is opened when the **Open editor automatically** checkbox is activated
  Alternatively, you can open the chart at any time by double-clicking on the chart symbol in the project navigator.

- When first creating a chart in a project, you will be prompted to import a block library.
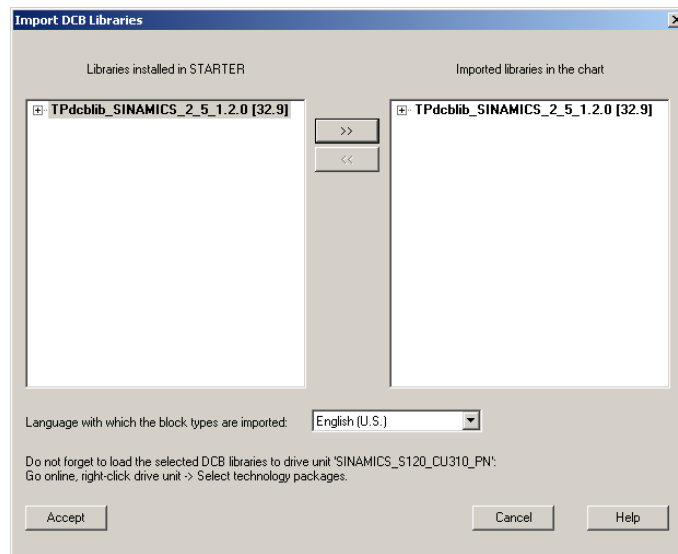
Figure 4-11   Importing a block library

- In the window **Import DCB Libraries**, select the block library in the left column **Libraries installed in STARTER.** Take the selected library into the right column by clicking the **>>** button.

- Close the window with the **Accept** command. The block library is loaded and the DCC editor opened with the chart.

The project structure has now been set up, a chart created and the block library loaded. All that remains is to create some activity within the chart, i.e. by inserting blocks and interconnecting them. Just one chart can be created for each drive object.

4.2.4 Inserting blocks

- Open a block family in the **Closed-loop control** family with the closed-loop control blocks.
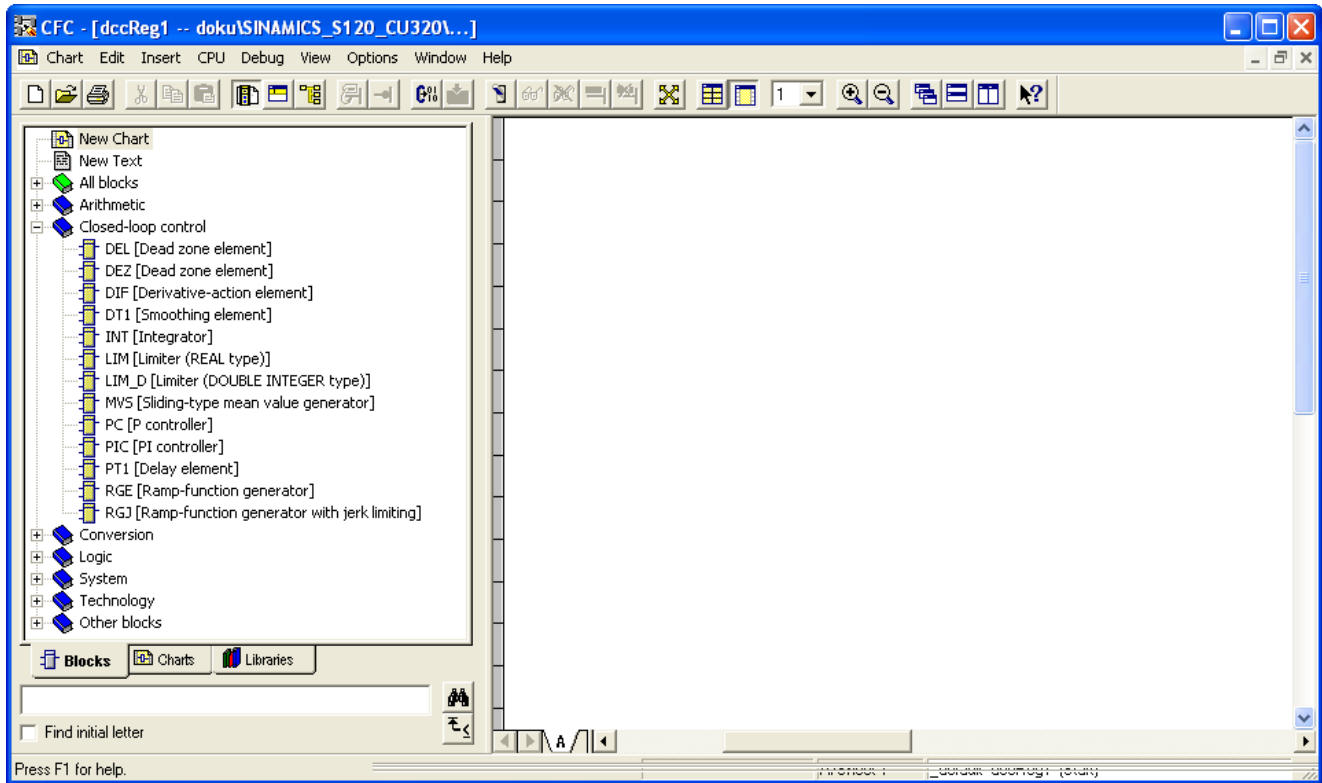
Figure 4-12   DCC editor with opened **Closed-Loop Control** DCB family

- Select the required block (e.g. INT) and insert it in the chart using drag-and-drop. Only the outline of the block in dashed lines is displayed during the copying procedure. Release the mouse button at the required point.

Note

If one block superimposes another block in the chart, the superimposed block will be displayed in gray and its connections will not be visible. You must reposition the blocks to ensure that all connections and block information are displayed.

4.2.5 Interconnecting blocks

Procedure

- Select the **Y** output of the first **INT**egrator, followed by the **X** input of the second **INT**egrator.
- Select the **Y** output of the second **INT**egrator, followed by the **X** input of the inverter (**SII**).
- Select the **Y** output of the inverter (**SII**), followed by the **X** input of the first **INT**egrator.

The autorouter creates the connecting lines from the outputs to the inputs and they are then interconnected.

4.2.6 Parameterizing block connections in the chart

For the first integrator INT 1/1, the following initial values should be assigned to the connections: LL = -10.0, LU = 10.0, SV = 2.0, Ti = 100 ms

For the second integrator INT 1/2, the following initial values should be assigned to the connections: LL = -10.0, LU = 10.0, Ti = 100 ms

To do this, open the **Properties - Connection** window of the respective block connection by double-clicking it. Enter the initial value for **Value** and click **OK** to close the window. Note that for the input of the values for Ti, the "ms" unit must follow the numeric value 100 without any spaces.

As the connections mentioned above are not interconnected, the entered values also remain valid after the initialization. The initialization of the block inputs and outputs is performed before the first cyclic call of the chart.
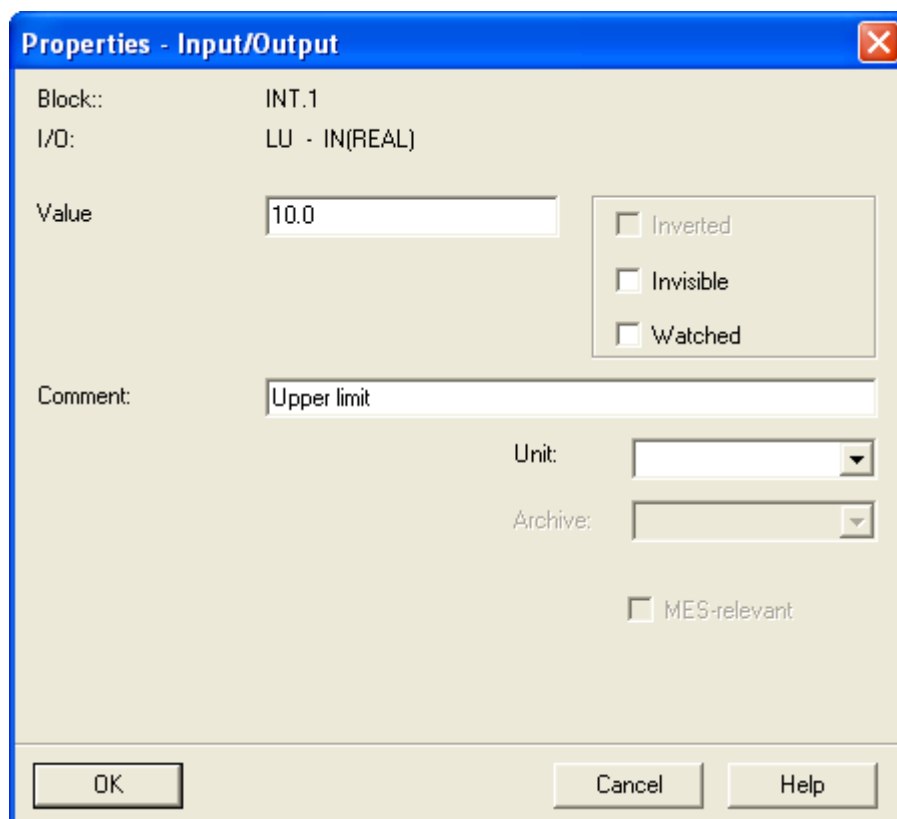
Figure 4-13   Properties window of the "LU" block connection
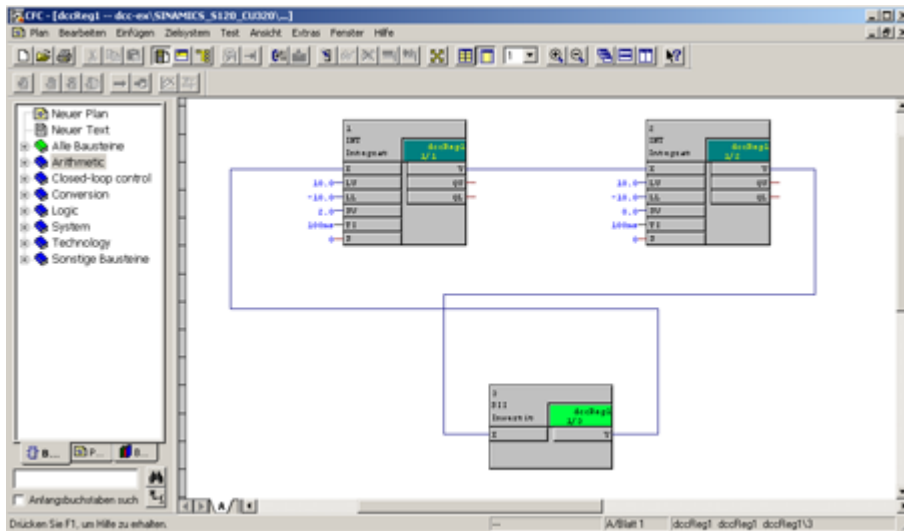
Figure 4-14   Chart "dccReg1" with interconnected blocks. Some connections have been assigned initial values that differ from the factory setting.

4.2.7   Publishing block connections as parameters

To be able to interconnect the output signal of the second (right-hand) integrator to the SINAMICS basic system, it must be published as an interconnectable parameters (i.e. as BICO parameter). To excite the oscillating circuit once, the integrator input of the first (left-hand) integrator must be published as adjustable parameter. The parameter numbers used in the following section have been chosen arbitrarily.

The Y connection of the second integrator is to be published as adjustable parameter. To do this, enter "@*20 output", for example, in the comment field in the Properties window (double-click connection Y). See Creating customer-specific parameters (Page 156).

Enter "@1 start" in the comment field of connection S (set) of the first integrator. This publishes the block connection as adjustable parameter.

As the default setting of the parameter number base of a chart is always 0, when the chart is compiled, the new parameters r21501 and r21520 are created and displayed in the expert list of the CU drive object in STARTER.

Note

To make this assignment of the @ parameters to the connections in the chart visible, the display form of the connections in the DCC editor must be changed. To do this, open the **Display Settings** window in **Options > Settings > Display**, change there under **Connections** the display from **Name** to **Comment** and click **OK** to close the window.

The chart will then be compiled.

Figure 4-15   **Display Settings** window

Figure 4-16   View from the expert list with the parameters defined in the DCC

4.2.8   Compiling the DCC chart in the DCC editor

Compiling
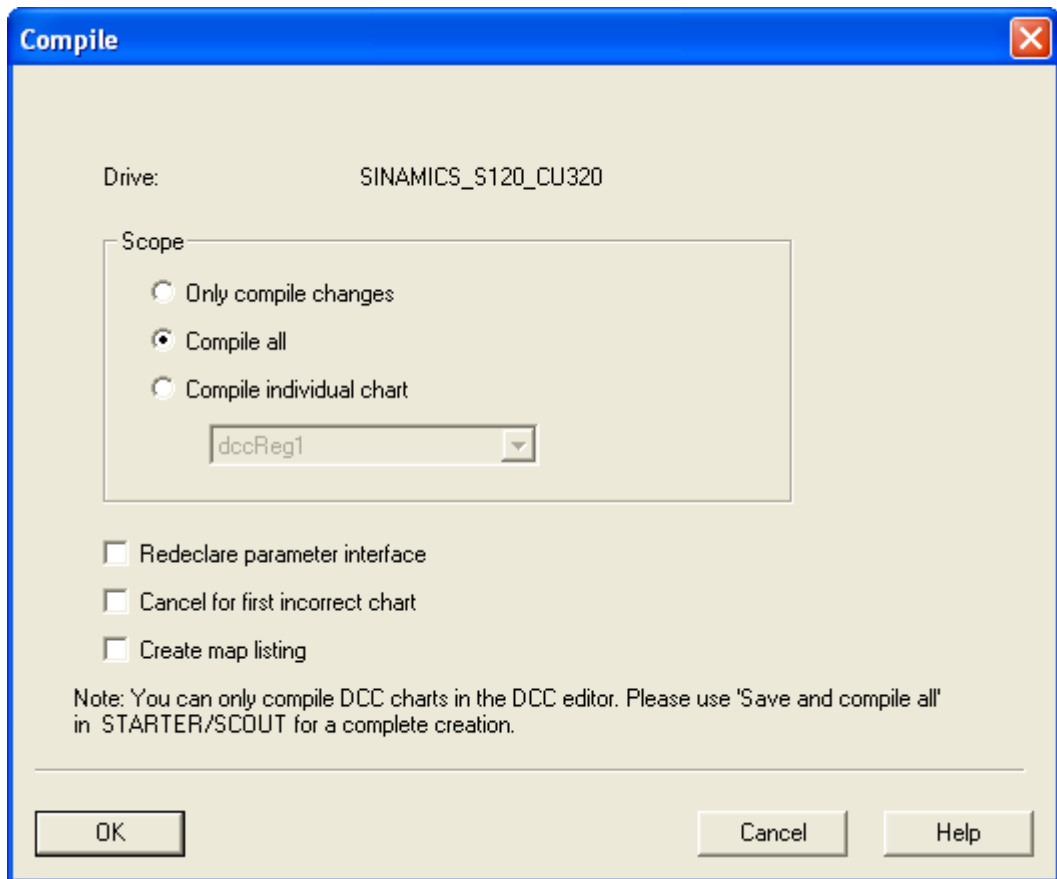
Figure 4-17   Compiling a dialog in the DCC editor

Before the first compilation from the DCC editor, the project must be saved once in STARTER (💾 button).

You can begin compiling from the DCC editor with **Chart > Compile > Charts as program...** or via the 🔳 button.

You can also start the compiling and saving of a project from the STARTER by clicking the 🔳 button.

If errors occur during compilation, the **Logs** dialog box will automatically be displayed at the end of the procedure (just as in the case of the consistency check).

Compilation options

For detailed information about the compilation options, refer to Compiling (Page 51).

After compilation

The compilation log is displayed after compilation. If error messages are displayed, the causes must be corrected before continuing.

Note

Once the DCC chart has been compiled, the interconnections from the DCC chart apply. For detailed information on accepting the interconnections from the expert list into the DCC chart, please refer to the section titled Reading back BICO interconnections and parameters (Page 188).

4.2.9 Setting execution sequence within an execution group

These are automatically assigned to execution group 1 when the blocks are inserted in a new chart. Execution group 1 is automatically given the name of the chart. The sequence corresponds to the order in which they are inserted in the chart. The sequence can be displayed and edited in the execution editor. You can access the execution editor in the DCC editor via **Edit > Execution sequence** or with ![icon]. If an execution group (light blue folder) is selected in the center column in the tree structure, the blocks contained in the execution group are displayed in the right-hand column.
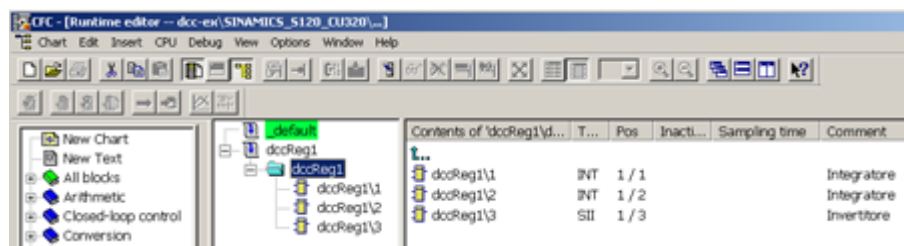
Figure 4-18   Execution editor with open execution group dccReg1

You can return to the DCC by selecting **Edit > Execution sequence** again or by clicking ![icon] again.

4.2.10 Setting sampling time for an execution group

The sampling time for an execution group is always set in the STARTER. The DCC you want is highlighted in the project navigator and the **Set execution groups** item called from the chart's context menu. The **Set Execution Groups** window opens.

Our chart was created on the CU drive object. r21002 = 4 ms in this case. A free execution group with a sampling time of 1*r21002 is set.

Note

In the offline mode of the STARTER / SCOUT V4.1.x, r21002 and r21003 are always displayed with the value. The r21002 value is identical with the p0115[0] value on the associated drive object.

When setting the sampling time, note that the minimum sampling time in the drive for DCC execution groups is 1 ms. Only times that are shorter than r21003 may be chosen as a multiple of r21002.

Please re-compile the chart once you have set the sampling time.

4.2.11 Loading the DCC technology option onto the CF card of the drive device

As of SINAMICS 4.3.1, the DCC technology package is located on the S120/S150 CompactFlash Card supplied from the factory for stand-alone drive units. With all other SINAMICS and SIMOTION D4xx CompactFlash Cards, the DCC technology package must be downloaded to the card by means of a technology package download

To load the technology package, call up SCOUT/STARTER and establish an online connection with the drive unit. SCOUT/STARTER is in online mode.

Then in the drive unit's context menu, select the **Select technology packages...** command.
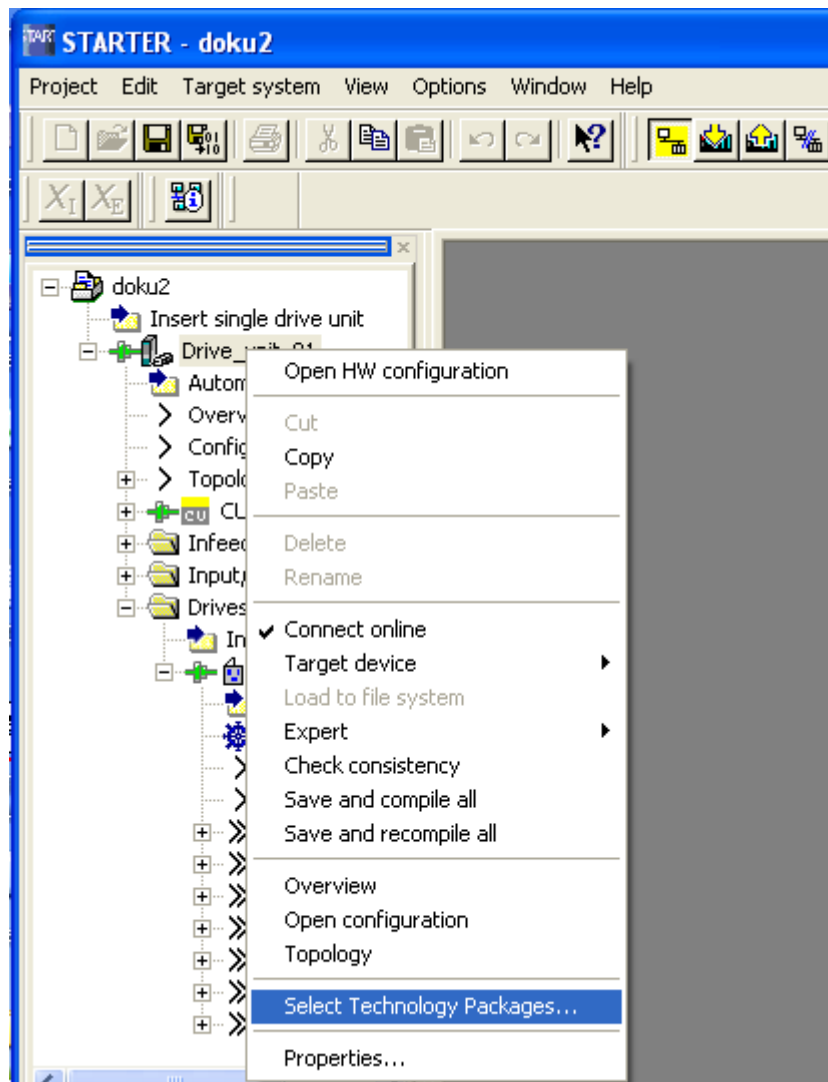
Figure 4-19   Context menu of the drive unit

In the **Select Technology Packages** window, select the desired block library and set **Download** for this library in the action column. Then click the **Perform actions** button in the bottom-right window above the progress bar. The label on the button changes to **Cancel**.
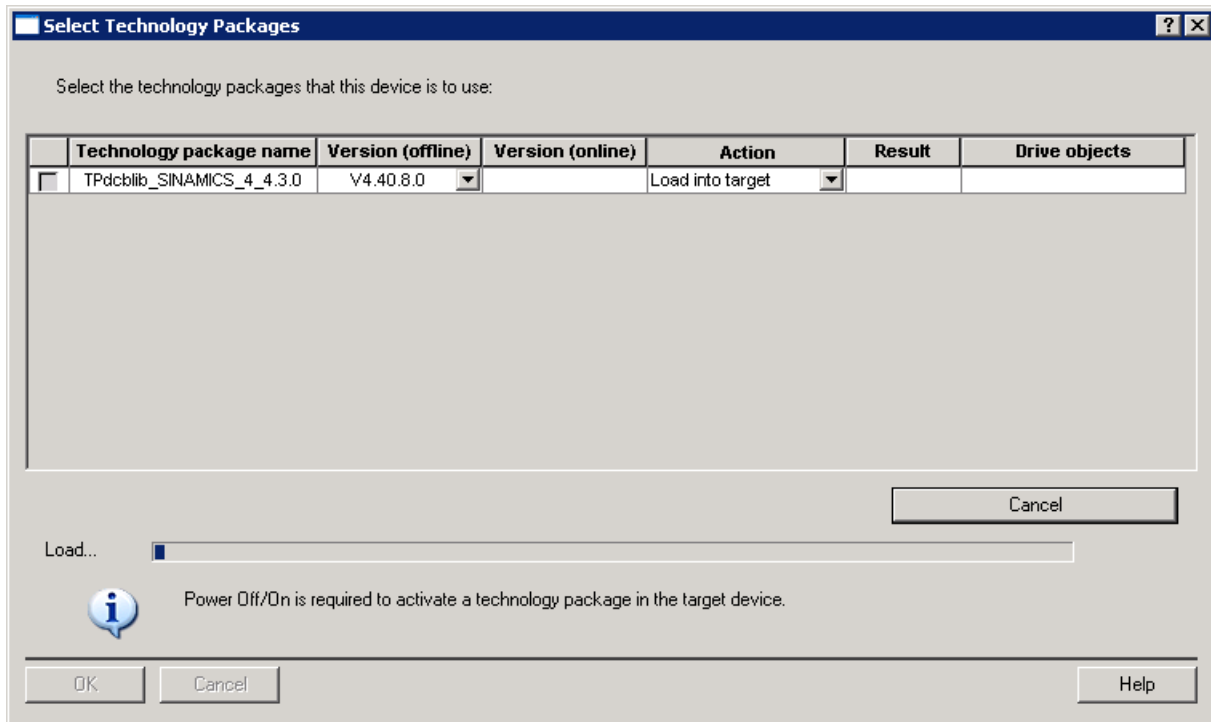
Figure 4-20   Select Technology Packages window

A progress bar shows how the downloading process is progressing. Once the downloading process is complete, the label on the button changes back to **Perform actions**.

The window is closed by double-clicking **OK**.

Note

Once the technology package has been downloaded, the drive unit has to be switched off and on again once. Only once the supply voltage to the CU and/or D4xx has been switched back on and it has been ramped up can DCCs be downloaded to the drive unit and run.

As of SINAMICS 4.4 and with the CU310-2 and CU320-2 modules, it is **not** necessary to switch off and restart the drive unit once the technology package has been downloaded. Once the technology package has been downloaded, an internal ramp-up process is performed automatically. DCC is loaded as part of this process.

When technology packages are deleted from the CompactFlash Card, all the DCC components of all the projects saved on it are deleted at the same time. As of SINAMICS V4.4 and with the CU310-2 and CU320-2 modules, an internal ramp-up process is then performed by the CU. This removes DCC from the CU memory.

Note

When downloading the technology package, as few people as possible should be connected to the bus, as the download time is significantly increased when multiple users are connected.

4.2.12   Downloading compiled DCC chart into the drive

To be able to execute the DCC program on a drive object, it must have first been downloaded to the drive unit. To do this, establish a connection to the drive unit and click the **Download** button in the Online/Offline Comparison window.

Downloading can take place in STARTER at any time via the **Download** function.

After the successful download, the DCC is calculated in cyclic operation on the CU drive object.

4.2.13   Displaying values of block connections online

As of SINAMICS V2.6 and SCOUT/STARTER V4.1.2, the DCC editor can display the values of block inputs and outputs online in test mode. The display is independent of whether a block input or output has been published as parameter. SCOUT/STARTER must be in online mode.

The signal value of block inputs (= connector inputs) published as BICO parameters cannot be displayed. Instead, the signal value of the supplying output can be displayed.

As soon as the test mode is exited again (e.g. by clicking), the yellow fields with the value display disappear.

If you want to display the values of the connections logged on once for display, by selecting the test mode again, you must set **Test > Laboratory mode** in the DCC editor. This setting is only possible when the DCC editor is not in test mode.

In our example, the output values Y of the two integrators are to be displayed. To do this, open successively in the DCC editor with a double-click on the connection of the **Properties – Connection** window and with a mouse click set a tick for **For test** on the right-hand side in the middle. This registers these outputs for display in test mode. Then use **Test > Test mode** or Ctrl-T or in the DCC editor to switch to test mode. The values of the two block outputs are highlighted in yellow and refreshed with a monitoring cycle of 2 seconds.
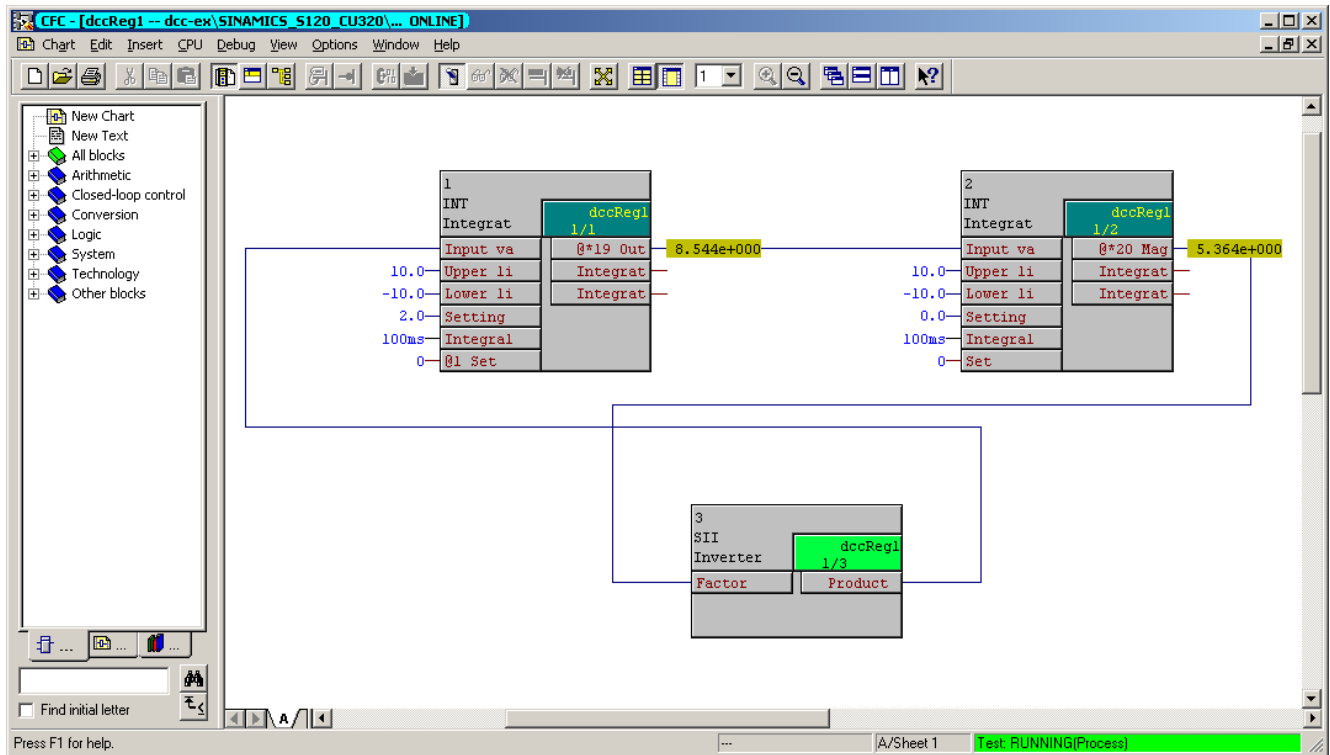
Figure 4-21   Display of the DCC dccReg1 in test mode

To make the oscillator vibrate, it must be excited once. To do this, set the p21501 parameter in the expert list of the STARTER once to "1" (the set value SV = 2.0 is applied at the output of the left-hand integrator) and then reset to "0". The oscillator vibrates and the output values of the integrators show changing values in the chart and in the expert list (r21520).

If you want to make further changes to the chart, you can exit the test mode with **Test > Test mode** or by clicking the ⬛ button. It can take several seconds before the test mode of the DCC editor is exited.

If you want to display the values of further block connections in test mode, right-click the desired block connection and select **Log on connection** in the context menu.
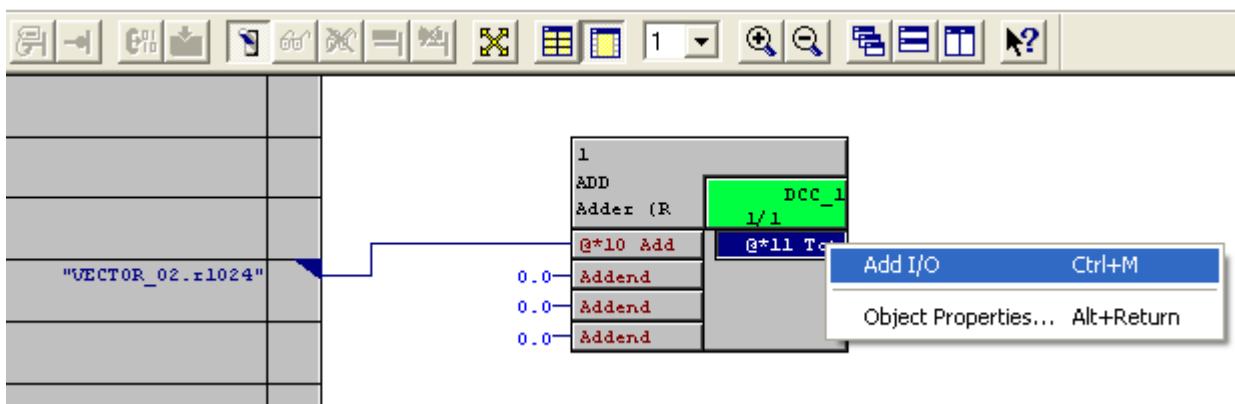
Figure 4-22   Log on connection context menu

If you want to display the signal value of a BICO output of the SINAMICS basic system, the signal value supplied to the DCC via the sheet bar cannot be displayed directly. In this case, an NOP_xx-block with the correct data type is inserted. The signal value of the output of the NOP_xx- can then be displayed online.

4.2.14 Interconnection to the BICO parameters of the basic system in DCC SINAMICS

Procedure

You can make an interconnection to the BICO parameters of the basic system as follows:

1. Open the DCC chart.

2. Select the DCB connection to be interconnected.

3. Publish the selected connection as a BICO parameter (the comment for the connection must start with "@*", see Creating customer-specific parameters ("publishing") (Page 156))

4. Right-click and select **Interconnection to operand...** from the context menu. The DCC Signal Selection window appears.

5. Select the parameter to be interconnected. Only those parameters are offered that are compatible according to the BICO interconnection rules. Further information on BICO interconnections can be found in the online help of STARTER/SCOUT under the BICO interconnections index entry.

6. Click OK to close the window.

The block connection is interconnected to the selected BICO parameter.

Note

The interconnected block connections must be published as BICO parameters for connections to signal outputs or signal inputs of the basic system.

The connection can first be made in the DCC editor without publishing the block connections. However, connecting an unpublished block connection to a BICO parameter of the basic system results in an error when compiling the chart.

Note

Interconnections with published block connections, which are established via the expert list, are not read back to the DCC editor. For this reason, subsequent compilation of the DCC chart will result in interconnections created previously in the expert list being overwritten again.

You can prevent interconnections from being overwritten by selecting **Chart -> Read back BICOs and parameters** in the open DCC chart in offline mode and then compiling the DCC chart.

Interconnection of different data types

When interconnecting to BICO parameters, the data types of the BICO parameter and pin may differ. In this case, you must ensure that the values are not outside the valid value range of the data types. Generally, the LIM DCC block should be connected between the input/output and the BICO parameter for this purpose.

If, for example, a block output with the REAL data type is connected to a BICO parameter of the UNSIGNED32 type, an exception error can occur when the value at the block output exceeds the value value range of an UNSIGNED32.

Deleting and moving BICO interconnections

When moving BICO interconnections, the signal can acquire the value 0 for a few cycles. The no longer interconnected input on the original connection is permanently assigned the value 0.

4.2.15 BICO interconnections and reading back parameters

To display up-to-date BICO interconnections, which have been changed online in the target device, in the DCC editor in online mode, the project must be read back from the target device.

It is also necessary to read back BICO interconnections and parameters to the DCC chart if these have been changed in the expert list. Unless these are read back, the configuration from/ in the DCC chart is reactivated after a subsequent download.

Procedure

BICO interconnections and parameter values changed subsequently can be read back from the target system via the DCC editor menu command **Chart -> Read back BICOs and parameters**. As of DCC Version 2.1, you can find this under **Options -> Read back BICOs and parameters**.

Input and output BICO interconnections are read back. The read back process always applies to all DCC charts of a device.

Requirements

● The structure of the block instances and interconnections must be the same both online and offline.

● The sources of the DCC chart must be available.

Note

It is not possible to read back BICO interconnections in the case of DCC charts that have been uploaded from a target device to an empty project.

4.2.16 Record with the trace signals from the DCC chart

The integrator output published as interconnectable parameter acts like any other connector output of the SINAMICS basic unit. This means this signal can also be recorded with the trace. The signal characteristic at block connections (that are not published as parameters) can still be recorded with the trace. The trace is called in STARTER with **Target system > Trace** or ![icon]. In this example, the outputs of the two integrators in the DCC are recorded with a cycle clock of 1 ms. The output of the first (left-hand) integrator is to be recorded as first signal in the trace. The signal selection is opened with ![icon] for this. Select the drive object on which the DCC is located (only CU_S_126). Open the directory tree and the dccReg1 chart is displayed. Open the directory tree of dccReg1 and the three blocks in the DCC are displayed. The desired connection is on block 1 (top left digit in the block symbol) of the dccReg1 chart. The designation of the connections in the signal selection of the trace is made up of the _ChartName_BlockNumber_, i.e. _dccReg1_1 in this case. The desired connection can now be selected in the list of block connections. The window is then closed by clicking **OK**. See the following figure.
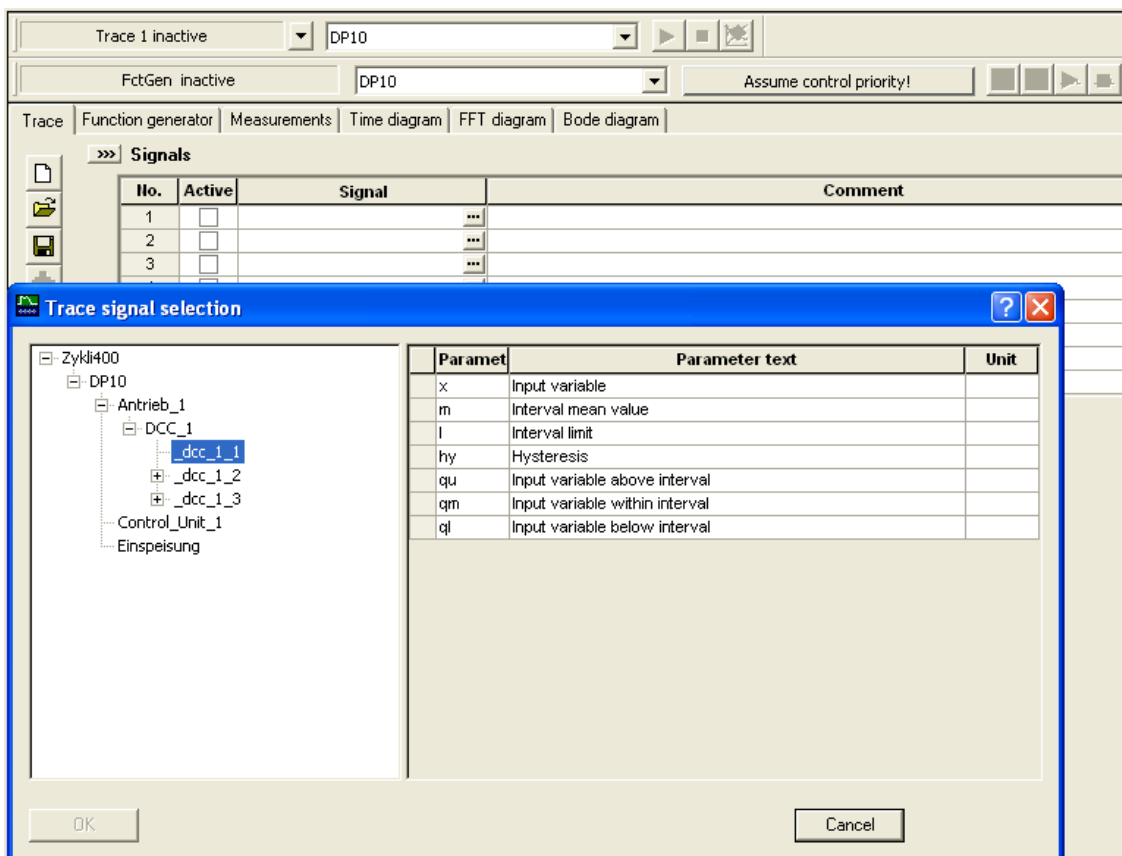
Figure 4-23   Trace signal selection for the block connections of the _dccreg1_1 block in the dccReg1 chart

The output of the second (right-hand) integrator is to be set as second signal. This block output is published as BICO parameter r21520. The recording can be started with the ![icon] button.
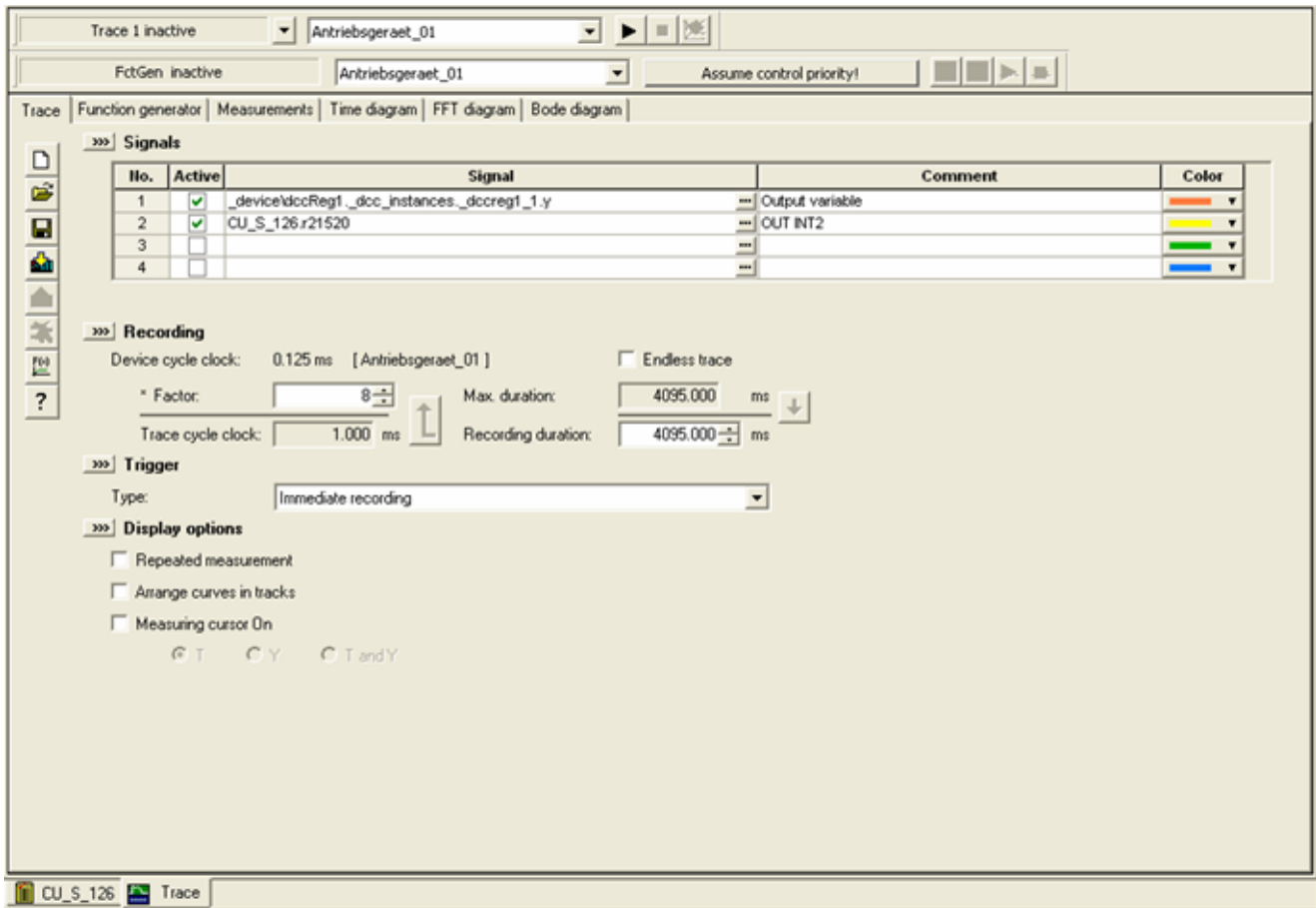
Figure 4-24   Parameterization of the trace
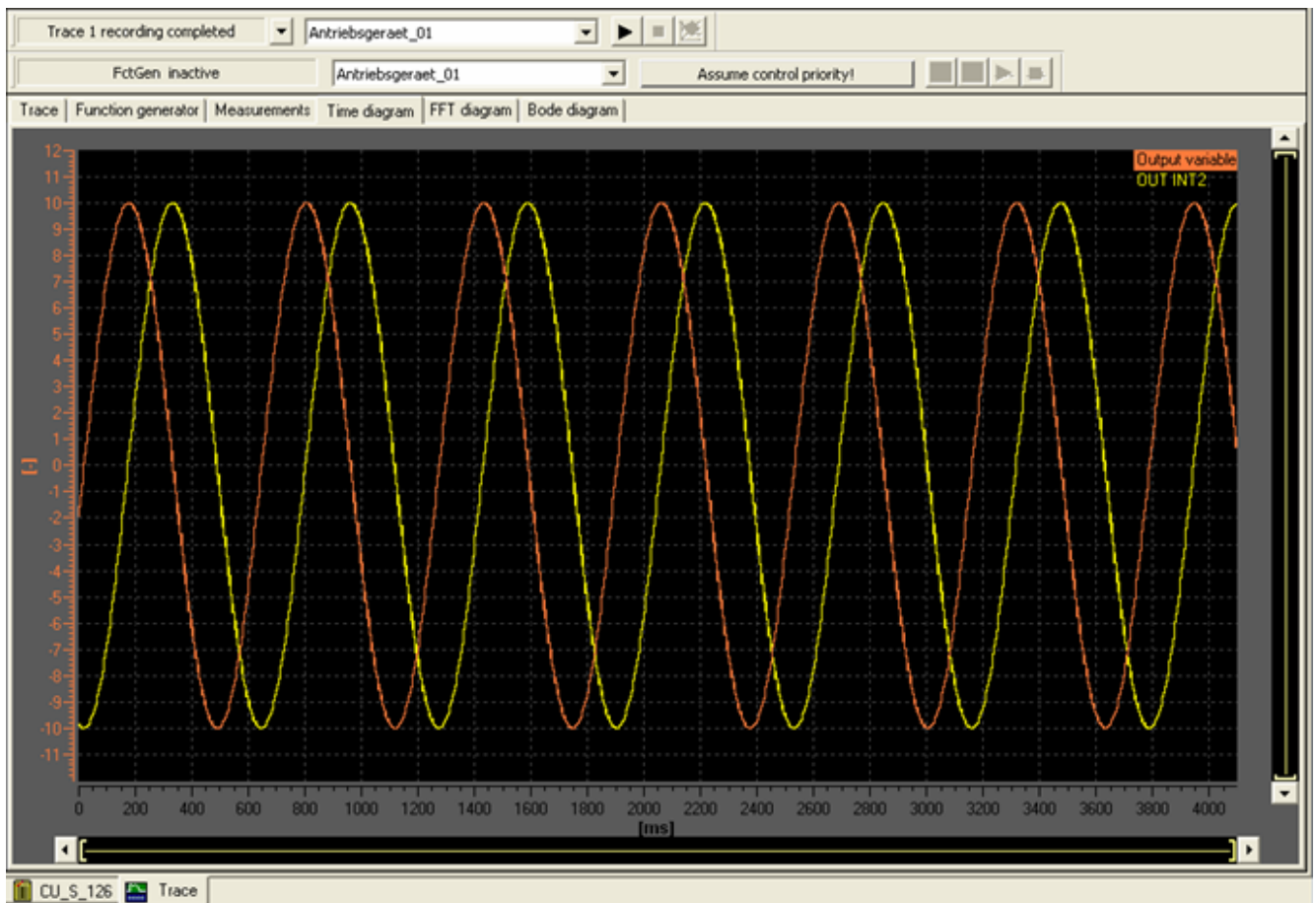
Figure 4-25   Display of the recorded signals as timing diagram

---

Note

When inserting block instances and connections online, the trace of the signal must be restarted.

4.2.17   Comparing DCC charts

For SINAMICS DCC charts on the drive object (DO) and in the DCC library, a detailed comparison of two DCC charts can be started in the project comparison as of STARTER V4.4 / SCOUT V4.4. The detailed comparison of the DCC charts can be started from the project comparison on the DCC chart.

The chart contents of two charts are compared on the basis of a chart source or compilation. The comparison result is displayed in tabular form. This feature is only available for SINAMICS DCC charts in SCOUT, not for SIMOTION DCC charts.

The detailed comparison can be started for the following comparison attributes:

- Source code (time stamp comparison)

- Blocks

- Interconnections

- Execution groups

- @ parameters and external references

The detailed comparison enables possible inconsistencies between the DCC charts to be precisely identified when going online or when activating the test mode.

The detailed comparison offers the following options:

- If the comparison attributes (source code, blocks, interconnections, execution groups, @parameters) are not consistent, a comparison result is provided with detailed information.

- Tabular list of the comparison result

- If the DCC charts are inconsistent, you can jump to the appropriate position in the DCC chart in order to avoid unnecessary searching in the DCC chart.

- As with the project comparison, the detailed comparison of DCC charts can be performed with the same comparison partners. This means that the detailed comparison can be performed between DCC charts and the project. The DCC charts and the project can be in the offline-offline or offline-online states respectively.

- The DCC libraries are also contained implicitly as "DCC chart" comparison object.

Further information on the DCC detailed comparison can be found in the Project Comparison Manual or in the STARTER/SCOUT online help (Project comparison index).

4.2.18   Archiving a project

If during or after the commissioning, you want to archive the project, including the DCCs, on a data carrier, this is possible with **Project > Archive** in STARTER.

4.2.19   Creating documentation

Complete documentation

The example has been successfully configured and tested. You can now create the complete documentation for your example.

Chart reference data

Alternatively, the chart reference data also displays the execution groups with the block types and the execution sequence. Use **Options > Chart reference data** in the DCC editor to open the **Chart Ref** window.

In the **Chart Ref** window, click the button to also display the blocks in the configured execution groups. To begin printing, click:
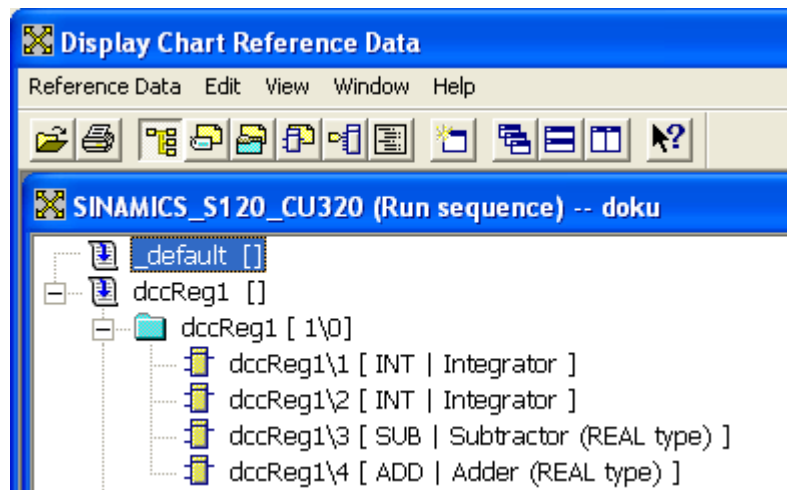
Figure 4-26   Chart Ref. window: Display chart reference data

Printing a chart

You should also print the chart to document the interconnection of the blocks. Because only one page was used on this chart, it suffices to print the current page. To begin printing, click:

If your chart consists of several pages, we recommend that you print them individually in the page view. Click the button to change to the page view.

Click or to customize the display so that all blocks are shown on a single page on the PC monitor.

Assignment of execution groups

To document the assignment of the execution groups, select the **Set execution groups** command in STARTER in the context menu of the chart. The **Set Execution Groups** window appears. Print a screenshot.
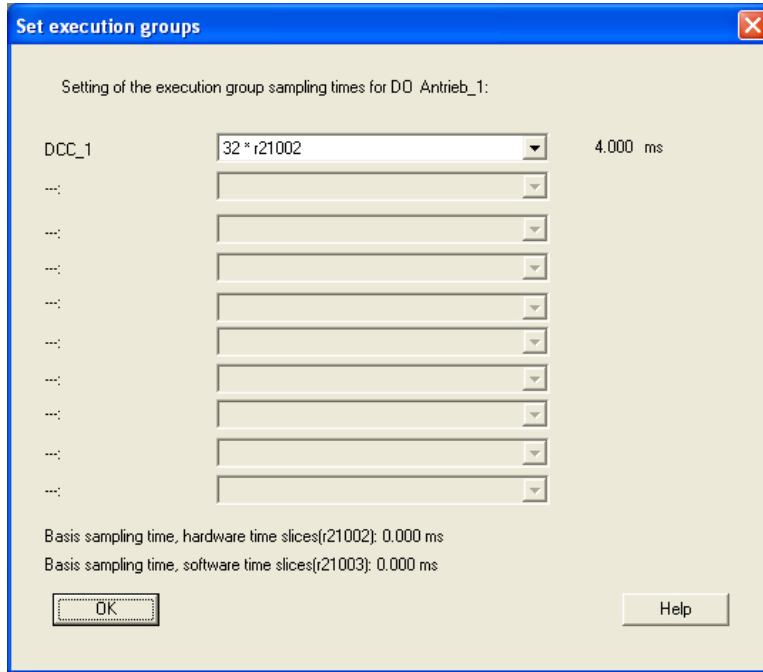
Figure 4-27   Set execution groups window

The change of the hardware r21002 basic sampling time displays the value for p0115[0] in the CU ramp-up of 1 ms. Here p0115[0] was changed from 4000 µs (factory setting) in offline mode to 1000 µs and then downloaded to the CU.

4.2.20   Update to a new SINAMICS version

When using DCC charts, the project must always be updated in the Engineering System and downloaded to the target device.

4.3 Connecting the DCC to the drive

4.3.1 Overview

Only DCB connections of a DCC chart declared as BICO parameters can be connected to the connector inputs (CI) and connector outputs (CO) of the drive.

All block connections of data type REAL that are published as BICO parameters are per-unit variables. This means that calculations are performed with per-unit signal values within DCC. (1.0 corresponds to 100%). The conversion to the connector units of the drive is performed automatically.

With all other data types, no conversion to a per-unit variable takes place.

4.3.2 Calculating a DCC chart with per-unit variables

Example 1.1 (interconnecting input value)

Disconnect any online connection to the drive unit in STARTER.
– p1020 = 1 and
– p1021 = p1022 = p1023 = 0
(function block diagram 2505) are set on a drive object of the SERVO (the "Extended setpoint channel" function module must be activated) or VECTOR type. Fixed speed setpoint 1 (p1001 in function block diagram 3010) is then output on r1024 (Speed setpoint active).

The following are also set:
– p1001 = 1500 rpm and
– p2000 = 3000 rpm (reference speed).

Insert a DCC on the drive object.

Insert the ADD block in this chart. Connections X1 and Y of the ADD block are published as (interconnectable) BICO parameters.

Right-click twice on input X1. The context menu opens. Select **Interconnection to operand…**. In the signal selection window DCC now select r1024 (CO: actual speed value effective) and confirm with **OK**.

Compile the chart by selecting **Chart > Compile > Charts as program…**.

Select Set execution groups… in the context menu of the DCC, set the only execution group that is present, e.g. to 20 * r21003 and confirm with **OK**.

Go online with STARTER and download the project to the target system.

Then online in the DCC editor, switch the test mode on with **Test > Set test mode**. Now right-click the output of the adder and select **Log on connection** in the context menu.
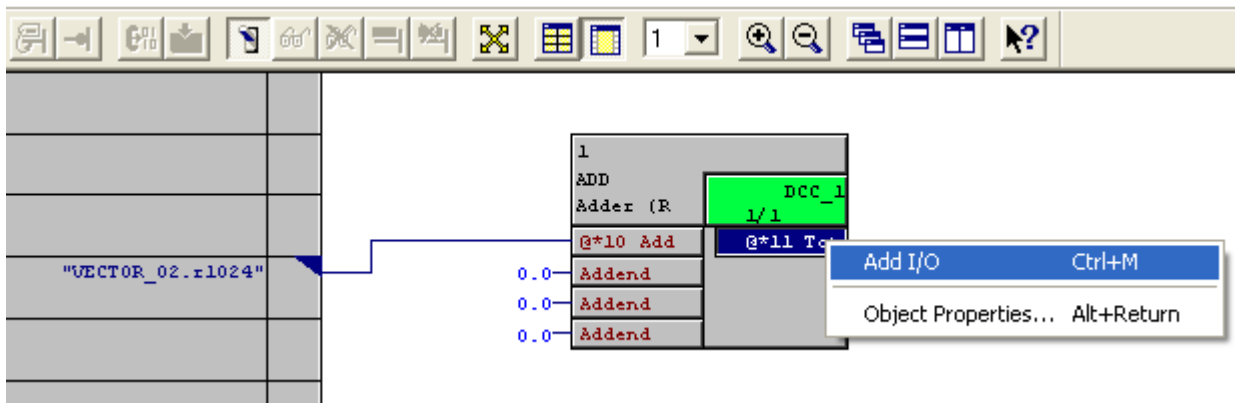
Figure 4-28   **Log on connection** context menu

The current output value (= per-unit speed setpoint = 1500 rpm / 3000 rpm = 0.5) is then displayed.
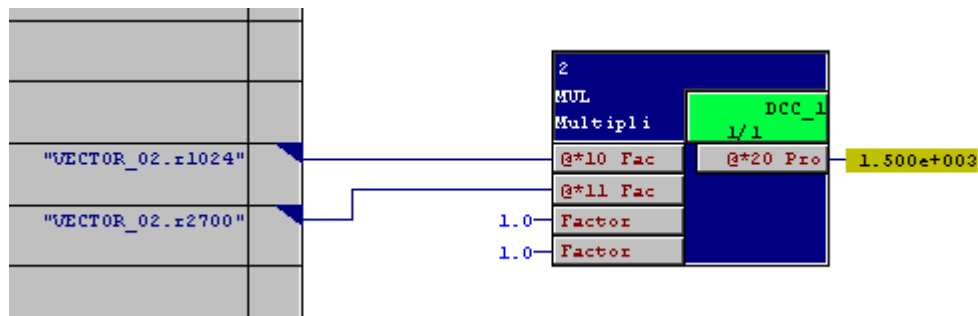
Figure 4-29   Connection value on example 1.1 is displayed

Example 1.2 (interconnecting the output value)

In the DCC chart, a value of e.g. 1.2 should be interconnected with the basic device. The value for this example will be configured as an initializing value to the X input of the NOP module. The module output (Y) has therefore the constant value 1.2, and will be configured as an interconnectable BICO parameter (@*20).

If a connector output of DCC is interconnected with the basic device, this will always be interpreted as a referred value of the basic system. The unit-related absolute value in the basic device now depends on which unit and therefore which reference variable the selected connector input has in the basic system:

When interconnecting the connector input for speed setpoint p1070 (function plan 3030), the referred output value from the DCC chart will automatically be multiplied with the reference variable "Reference speed" p2000 (in the example p2000 = 1500 min^{-1}). In the SINAMICS basic system, a speed setpoint of 1800 min-1 (= p2000 * 1.2 = 1500 min^{-1} * 1.2) will thus become effective.

For this reason, always take note in the case of BICO interconnections with the basic system if you work with absolute variables or with referred variables in DCC. Since the connector output of the DCC chart in the case of interconnection with a unit-related connector input of

the basic system will always (internally) be multiplied with the reference variable (belonging to the unit).

4.3.3 Calculating a DCC chart with absolute variables

4.3.3.1 Example 2.1 (interconnecting input value)

If you want to work with absolute variables in the DCC (in our example using speed values), you have to convert the referenced value to which the variables are automatically converted by the drive into an absolute variable (in our example, a speed) using the reference variable. For DCC, the reference variables are provided as monitoring parameters r2700… r2707 which have no units and can be interconnected. Parameters r2700… r2707 have the special feature that the value is transferred 1:1 to DCC without being divided by the reference variable.

On SERVO type (the "Extended setpoint channel" function module must be activated) or VECTOR type drive objects,
– p1020 = 1 and

– p1021 = p1022 = p1023 = 0
(function block diagram 2505) are set. Fixed speed setpoint 1 (p1001 in function block diagram 3010) is then output on r1024 (speed setpoint active, function block diagram 3010).

The following are also set:
– p1001 = 1500 rpm and
– p2000 = 3000 rpm

r1024 (speed setpoint active, function block diagram 3010) then assumes the value 1500 rpm.

The interconnectable monitoring parameter for the r2700 reference speed indicates a value of 3000 in the expert system.

The multiplier MUL is dragged into the chart.

The 1st factor (X1) is published as a BICO parameter and interconnected with r1024. → Factor 1 = 0.5 (referenced speed setpoint active)

The 2nd factor (X2) is published as a BICO parameter and interconnected with r2700. → Factor 2 = 3000.0 (value of reference variable p2000)

At the multiplier output this results in:

0.5 * 3000.0 = 1500.0

The speed setpoint r1024 is available again in DCC in rpm.
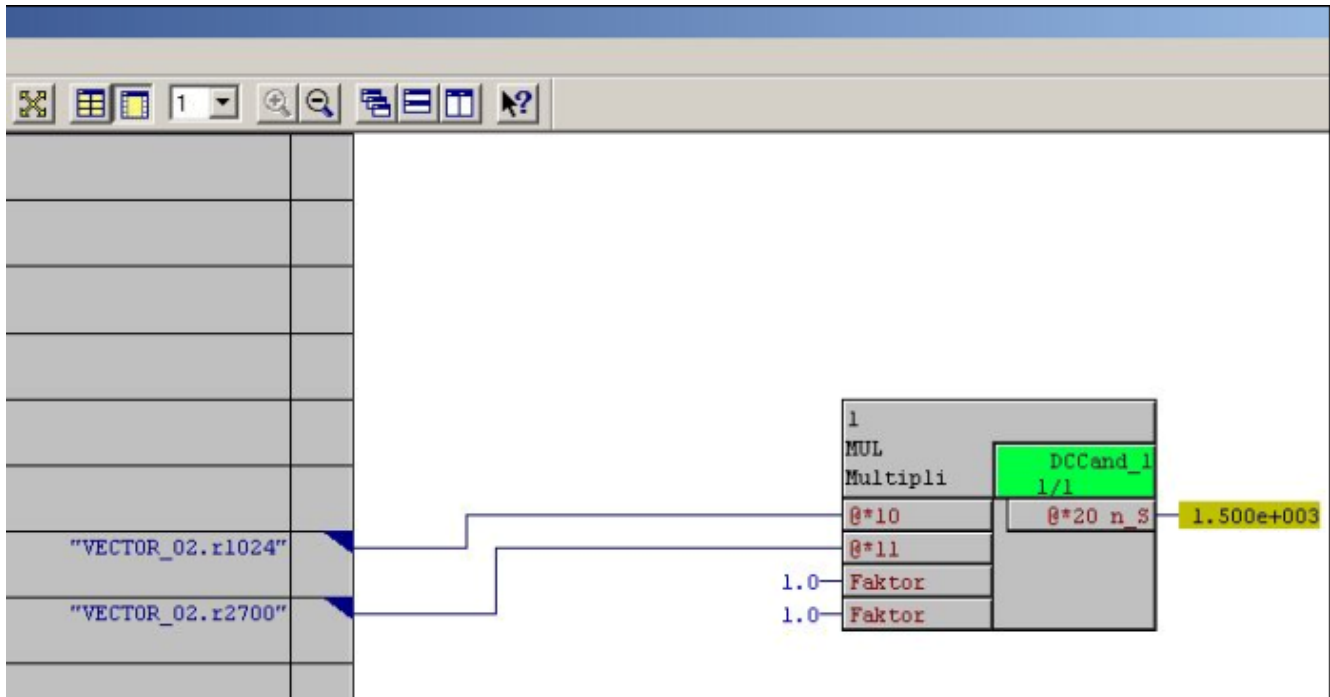
Figure 4-30   DCC for example 2.1 in online test mode

The block was automatically assigned to an execution group when inserted into the chart. The free execution group 20 * r21003 (selected at random) has been assigned to the execution group in the **Set execution group** menu.

4.3.3.2 Example 2.2 (interconnecting output value)

Absolute values are used in the DCC to calculate the torque in Nm units. The torque setpoint calculated is to be passed on to the "Extra torque M_extra 2" (p1513, function block diagram 6060) connector input. For this to happen, the absolute value must be converted into a referenced torque while still in the DCC. When connecting a real type (== floating point) connector output (CO) from the DCC with connector input (CI) p1513 M_extra 2 [torque in the Nm unit] on the drive, the automatic conversion in the BICO connection always assumes that DCC is providing a referenced signal.

Note

The LIM type block supplies the absolute torque of 0.204625 Nm calculated in the chart at its Y output. This absolute value is now divided by the reference variable for torque r2703 = 0.8185 Nm and the referenced torque value for interconnecting with the drive is thereby calculated. The referenced torque value is available at the block DIV's output and is published as BICO parameter r21530 (value = 0.2499 = 0.204625/0.8185). Block NOP1 has only been inserted so that the value of reference torque r2703 can be shown in the chart at its r21527 output published as a BICO parameter.

Monitoring parameter r1515 (total extra torque, function block diagram 6060) is only calculated if the speed controller is enabled and the drive is magnetized (r0056.4 = 1 = yes). (A machine must be connected to the motor modules and be running for this purpose.) Otherwise the value is set to zero.
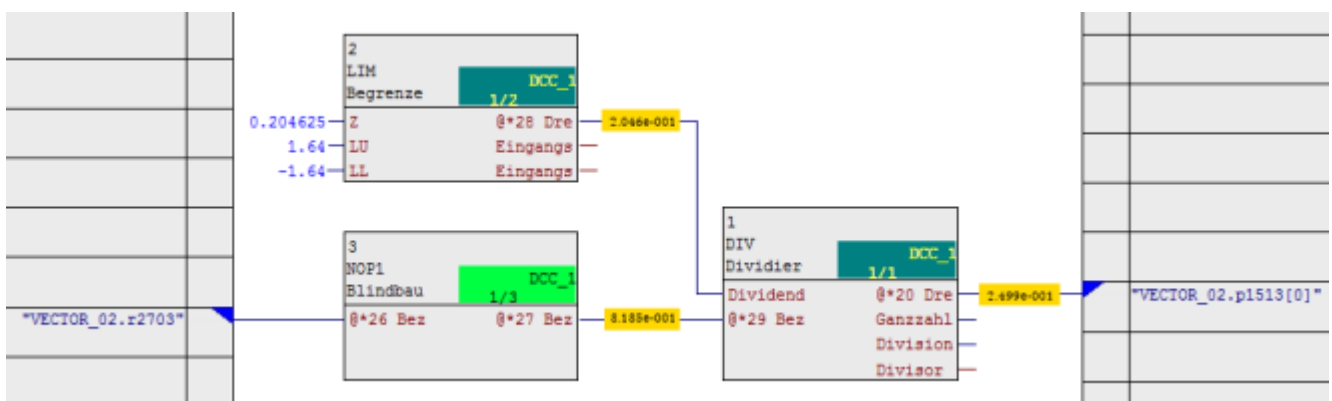
Figure 4-31   Interconnecting output value

4.3.4 Interconnecting DCC signals with communication interfaces IF1 and IF2

4.3.4.1 Preliminary remark

The basic system can be connected to process data interfaces IF1 and IF2 via free telegram configuration using BICO (p0922 = 999) or via (standard) telegrams depending on p0922. To interconnect with the basic system, the DCC connections needed must be published as BICO parameters. These DCC parameters should always be interconnected with the basic system in the DCC Editor by calling the context menu **Interconnection to operand**.

4.3.4.2 Interconnecting received process data with DCC

When interconnecting the received PZD data (see SINAMICS S Parameter Manual, function block diagrams 9206 and 9204), the particular way in which the connector outputs (COs) of PZD processing behave (for IF1 r2050, r2060 and for IF2 r8850 and r8860) should be noted. These COs (connector outputs) can either provide their signals in whole numbers (integers) or as floating point values (REAL). The data type provided is determined by the first signal linked to this CO. This can also be determined by a standard message frame having been selected (p0922 |= 999) on a drive object. The PZD COs are thereby automatically interconnected according to the definition of the message frame selected. This interconnection is not canceled when resetting p0922 = 999 == free message frame configuration with BICO.

p1070 = 2060[2] is set for example. r2060[2] (PZD received word 3 and 4) is thereby connected to the main setpoint (in function block diagram 3030) of a REAL variable of the basic system. This means that only connector inputs from DCC with the REAL signal data type can be interconnected on r2060[2].

Note

Connections to integer inputs can be made both online and offline in the DCC editor; a corresponding error message is only issued when the chart is downloaded.

4.3.4.3 Interconnecting sent process data with DCC

The send data is interconnected as with any other BICO connection.

Detailed descriptions about this topic are contained in the *SINAMICS S120 function manual.*

4.4 DCC SINAMICS specifications

4.4.1 Rules for assigning names in the DCC editor

Names are used for data exchange between SCOUT/STARTER and the DCC editor.

The names in the DCC editor must therefore abide by the following rules:

- Basic chart:
  - No keyword or previously defined name permitted
  - Must start with a letter or underscore
  - May contain numbers, letters, and underscores
  - An underscore must be followed by a number

- Subchart:
  - Must start with a letter
  - May contain numbers and letters
  - Underscores are not permitted
  - Keyword or previously defined name permitted
  - In the DCC editor, a check is performed to determine whether the name has previously been defined.

- Block instance:
  - Can start with a number
  - May contain numbers and letters
  - May not start with an underscore
  - Keyword or previously defined name permitted
  - In the DCC editor, a check is performed to determine whether the name has previously been defined.

- Execution group:
  - Keyword or previously defined name permitted
  - In the DCC editor, a check is performed to determine whether the name has previously been defined.

- BICO parameter:
  - No previously defined parameter numbers permitted
  - In the DCC editor, a check is performed to determine whether the parameter number has previously been defined.

4.4.2 Field/name lengths and conventions

Field/name lengths and conventions

Object	Length	Remark
Chart name	22 *)	May not contain the following characters: \ . : / * ? " < > | # % ( ) Use of the "_" character is subject to particular specifications.
Chart comment	255	All ANSI characters are permitted.
Execution group	22	Only letters, numbers and underscores may be used.
Block instance comment	80	All ANSI characters are permitted.
Parameter comment	80	All ANSI characters are permitted.
Block name (instance name)	16 *)	May not contain the following characters: \ . : / * ? " < > | # % Use of the "_" character is subject to particular specifications.
Global operand	Max. 49	Interconnection with BICO parameter.

*) The chart name and block name must jointly consist of a maximum of 24 characters, including separating characters.

4.4.3 Representation of the dynamic value display

The values are output next to the connections, according to their data type. They are displayed on the screen with a colored background.

Table 4-2   Representation of the dynamic value display

Representation	Meaning
Blue on white	Representation of the values in edit mode (offline)
Black asterisks on yellow	Values during transfer to the dynamic display
Black value on yellow	Representation of the values read from the drive object in test mode
#### on a red background	Representation of values while the dynamized values required from the drive object are missing (fault, overload)

DCC for SIMOTION

5

5.1 Overview

5.1.1 Introduction

This product brief is intended for experienced SIMOTION users who are not yet familiar with the DCC program package. Using a short example, you will find out how to create a project after starting up SIMOTION SCOUT, create a DCC chart, interconnect blocks, compile the chart, download it to the target device and test it online.

Note
In most cases, there are a number of options for working with the DCC editor (e.g. using the keyboard). In this example, the quickest or most suitable option is used. Apart from a few exceptions, no alternative methods of operation or procedures are explored here.

5.1.2 Software requirements

The software requirements for DCC are the same as those for SIMOTION SCOUT.

You require an appropriate license to use the DCC editor.

Please refer to the following table for information on which kernel version supports which DCB libraries.

Kernel version	DCB lib version
SIMOTION 4.1.5	4.1.2, 4.1.4, 4.1.5
SIMOTION 4.2	4.1.5, 4.2

5.1.3 SIMOTION system integration

5.1.3.1 Execution level, execution group and execution sequence

DCC tasks and execution groups
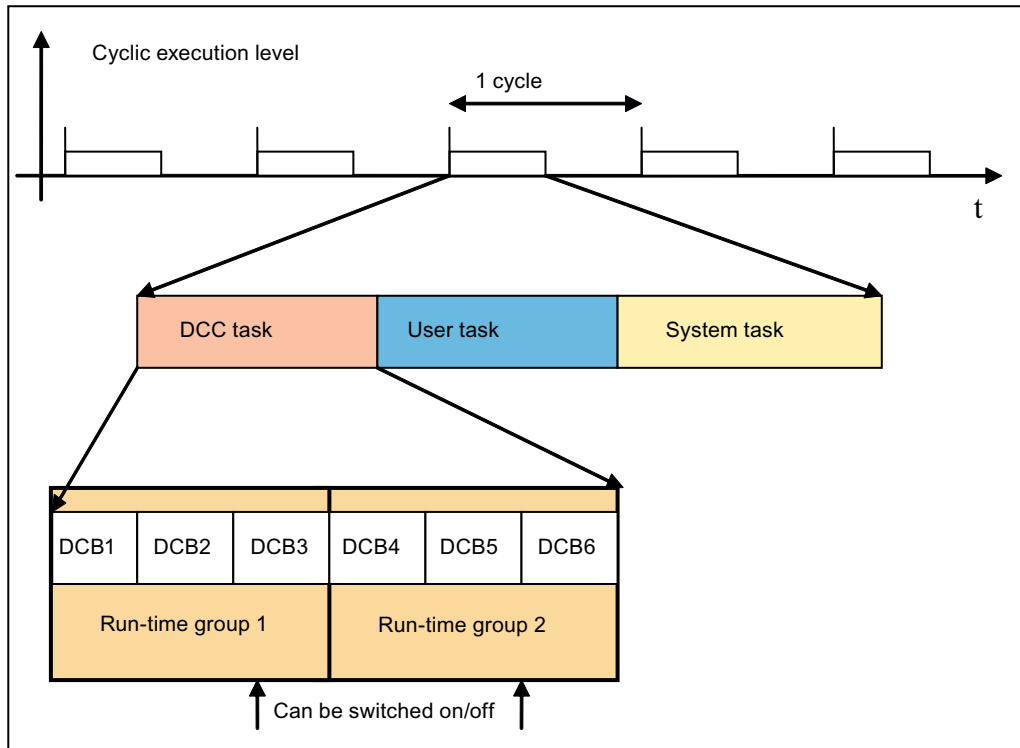
Figure 5-1   Sequence model for blocks

Tasks

A DCC task is available for the user programming in each of the five cyclic execution levels (in descending order of priority):

- T1: servodcc in the servo level

- T2: ipodcc in the IPO level

- T3: ipodcc_2 in the IPO_2 level

- T4: dccaux in the DCCAUX level

- T5: dccaux_2 in the DCCAUX_2 level

Further information on the execution system of SIMOTION can be found in Section *Execution system / tasks / system cycle clocks* of the *SIMOTION SCOUT Basic Functions* Function Manual.

Execution groups

The available tasks (sampling time and execution sequence in the system: T1 to T5) are assigned execution groups. The blocks are embedded into these execution groups. The execution groups therefore allow the task to be structured or divided as required by the user; e.g. dancer control, setpoint processing, etc. The execution groups are embedded sequentially into the blocks. **Only** blocks **from one** basic chart may be contained in an execution group.

You can use the "Enable" attribute to activate and deactivate an execution group. Normally, execution groups are processed in cycles; however, the enable attribute can be used for switching individual execution groups on and off.

A BOOL-type block output can be connected to enable an execution group or a block group. To do this, highlight the connection to be interconnected and select the **Interconnection with execution group**... command in the context menu.

Note

The **Interconnection with execution group**... command is only supported with DCC SIMOTION.

Execution sequence

For tasks: The execution sequence corresponds to the sequence in which execution groups and blocks are inserted within a task.

The same applies to groups: The sequence in which the individual blocks are inserted is the execution sequence within the execution group.

You can change the execution sequence.

Setting the system cycle clocks

The properties that determine the execution behavior of a task are configured in SIMOTION. You can carry out parameterization in the Properties dialog box of the CPU under **Expert execution system > Set system cycle clocks**.

No multiple insertion of blocks in different tasks

Inserting a block several times in different tasks is not permissible.

Changing the execution sequence

- Click **Edit > Execution sequence** or the ▣ button.

The **Execution Sequence** window appears: In the left-hand side of the window, you will see the structure of the task. In the right-hand window, the content is displayed. The default integration position of the blocks is in cyclic task **T2**, in the default execution group that has the same name as the chart.

To move blocks that have already been inserted from task **T2** to higher-priority task **T1**:

- Double-click the **T2** symbol.

- The execution groups are displayed in the right-hand window:

  – When you highlight the execution group, the blocks it contains are displayed (including the chart/block name, comment and position).

  – Highlight the execution group that you want to move in task **T2** and use drag-and-drop to drag it to the task **T1** symbol in the left-hand window. The right-hand window is now empty.

  – When you double-click task **T1** to open it, the execution group that you have moved will be displayed there.

- Close the window via **Chart > Close** and proceed to the next step.

5.1.3.2 HMI variables (publishing of variables and @ variables)

HMI variables

You can declare block inputs and outputs as **HMI variables** and therefore generate a static interface for these for use in your system visualization. This interface is largely static from an HMI point of view, i.e. not every change made to the DCC configuration demands that the address information of the HMI system be reimported.

The memory image with the HMI variables is **not** automatically deleted during compilation. It is only deleted on request.

HMI variables that are removed from the DCC when the block that defined them is deleted remain in the memory image until **reorganization** is requested during compilation in the DCC.
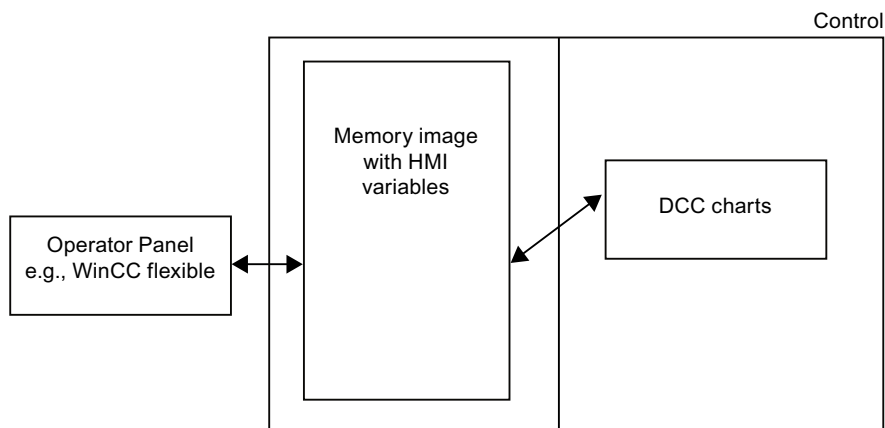
Figure 5-2   HMI variables

@ variables (SIMOTION)

Structure of the @ variables

In the comment, block inputs and outputs are published by the variable name preceded by @.

Table 5-1   Structure of comments in SIMOTION

Comment	Meaning
SIMOTION: @name <<variable comment>>	The connection is entered as an HMI variable in SIMOTION. Name is the part before the first blank (whereby you must adhere to the ST conventions (Page 228)).

The text which follows the variable name (separated by a blank), is transferred as a variable comment. It is then displayed in the symbol browser. The variable identifier results from the variable name.

The data type of the published input/output is taken over by the block connection. Where necessary, it is mapped onto the appropriate data type of the engineering system.

No help function can be created for @ variables.

Exporting to WinCC

Procedure

A created DCC chart can be exported to WinCC, whereby the export specifications of the SIMOTION SCOUT also apply for DCC charts. However, for this to happen the OPC-XML export must be activated in the Properties dialog box of the chart in SIMOTION SCOUT.

In the engineering system, right-click the chart and select **Properties** in the context menu. The export is activated via **Enable OPC-XML** on the **Compiler** tab.
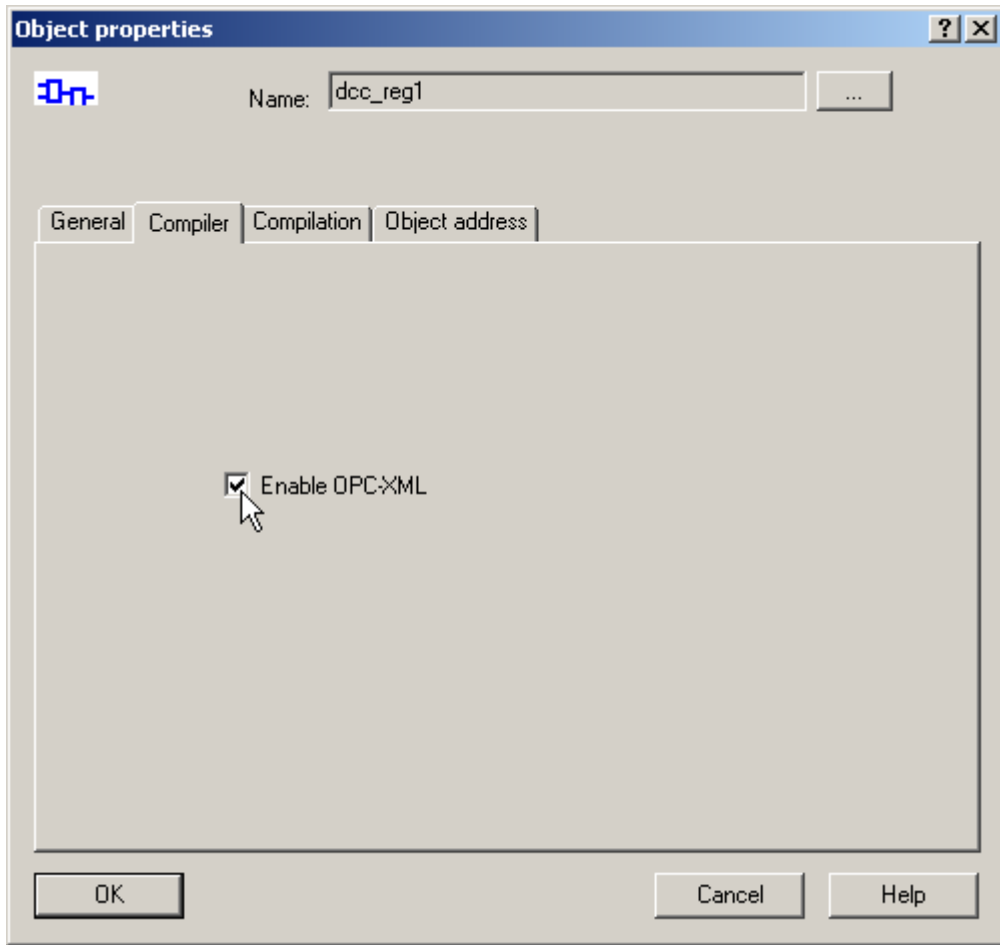
Figure 5-3   DCC chart properties - OPC-XML export

The export can be performed in SIMOTION SCOUT via **Options > Export OPC data**.

5.1.3.3   Interconnecting with SIMOTION variables

Interconnecting with SIMOTION variables

You can interconnect with global operands (SIMOTION variables) in the DCC editor.

Note

Please observe the Specifications for assigning names in DCC SIMOTION (Page 228).

Procedure

Interconnection can be performed as follows:

1. Select the block connection to be interconnected.

2. Execute the **Insert > Interconnection to operand** command in the menu. The interconnection to a global operand can also be performed via the context menu of the block. The **Symbol Input Help** window appears.
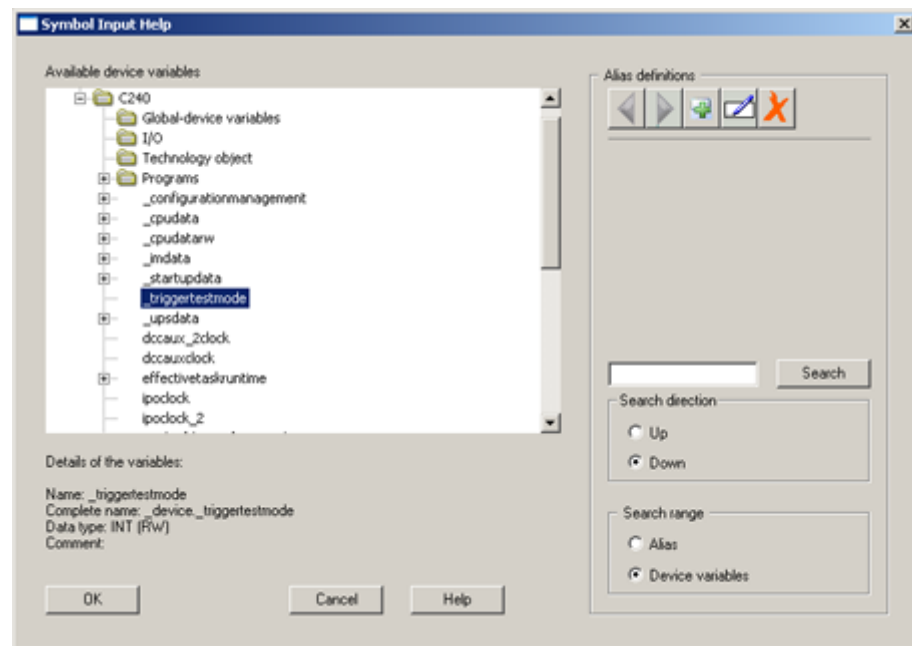
Figure 5-4   **Symbol Input Help** window

3. You can now navigate in the CPU or technology object assigned to the chart and select the device variable to be interconnected – the application highlights compatible device variables in bold-face type.

4. Click **OK** to close the window.

From this point on, the selected block connection is interconnected with the global operand and the global operand is represented by a sheet bar variable in the DCC editor.

Note

The global variables and the I/O variables must first be created and then the interconnection performed.

Note

You can search for a variable name or a parameter text in the search field of the **Symbol Input Help** dialog box.

Interconnection to array elements

You can interconnect array elements of ST programs in the DCC editor.

Procedure

Interconnection can be performed as follows:

1. Create a new ST program with an array variable in the SIMOTION SCOUT engineering system using the **Insert ST program** command.

2. Right-click and select **Interconnection with operand** from the context menu. The **Symbol Input Help** window appears.
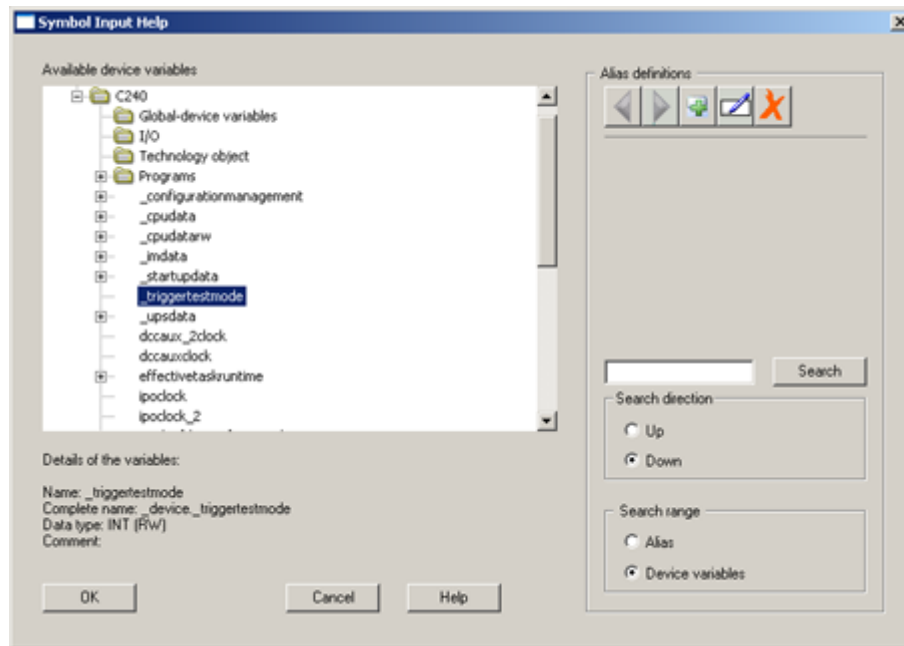
Figure 5-5   Symbol input help window

3. Select the array variable to be interconnected.

4. Enter the number of the array element to be interconnected.

5. Click **OK** to close the window.

From this point on, the selected block connection is interconnected with the array element and the array element is represented by a sheet bar variable in the DCC editor.

Alias parameter

You can assign alias identifiers in the DCC editor for variables. The reason for assigning an alias may be that a name is too long - the DCC editor only accepts a maximum of 49 characters for identifiers.

Procedure

An alias can be created as follows:

1. Right-click the block connection and select **Interconnection with operand** from the context menu. The **Symbol Input Help** window appears.

2. Select the connection that is to be defined by the alias.

3. Click the button from the alias definition area.

4. Enter the alias name in the entry field.

5. Close the window.

The alias has been successfully created and can now be used - the computer now resolves the alias as required, e.g. within the framework of the compilation.

5.1.4 DCC and SIMOTION trace

As accustomed with SIMOTION-based devices, you can trace each block connection as a variable or use it in the trigger condition in SCOUT.

5.2 Working with DCC SIMOTION

5.2.1 Preliminary remarks on configuration

Preliminary remarks

The following is a brief explanation of what you will be configuring in this chart.

Configuration example

The configuration example deals with a straightforward oscillating circuit that creates sinusoidal oscillation at its output.

It will only take you a few minutes to create the chart; then you can execute it in test mode as a demonstration.

The following blocks are used:

- Two integrators (**INT**)

- One inverter (**SII**)

As indicated by the differential equation $f''(x) = -f(x)$, the oscillating circuit is comprised of two integrator blocks that are linked by negation.

The frequency of the oscillating circuit is determined by the integrating time constant at the integrators.

The oscillation amplitude is specified by the initial value at the integrator output.

The DCC editor can be started via the SIMOTION SCOUT engineering system.

Configuration example structure

The configuration example is divided into the following steps:

1. Creating a new project

2. Inserting the DCC in the project

3. Inserting blocks in the DCC

4. Interconnecting blocks in the DCC

5. Compiling the DCC in the DCC editor

6. Downloading the compiled DCC program to a CPU

7. Starting the CPU

8. Switching between process and laboratory mode

9. Setting test mode and the test mode types

10. Monitoring values in laboratory and process modes

11. Logging on and logging off connections for monitoring

12. Using the "Enable" attribute for execution group control

13. Creating complete documentation for the example

14. Information on the chart reference data function

5.2.2 Creating a project

- Create a new project in the SIMOTION SCOUT engineering system, e.g. **dcc_ex**, see Creating a project (Page 23).

- Create a new device, e.g. a D445, using **Create new device**.

- Select the package **TPdcblib_SIMOTION_4_2.3.0 [x.y]** and close the window.

- You can now insert a chart.

5.2.3 Inserting a chart

- In the project navigator, double-click **Insert DCC** under **Programs**.

- Now give your chart another new name. The text field for the DCC name is selected automatically and the cursor is activated.

- Edit the new name here: **dcc_reg1**.

- If the **Open editor automatically** option is activated, the DCC will be started automatically. Alternatively, you can open the DCC by double-clicking **dcc_reg1**.

- If a library has not yet been imported, at this point you will be prompted to import one: Select **TPdcblib_SIMOTION_4_2.3.0 [x.y]** in the left-hand window, then click **>>** and finally **Close**.
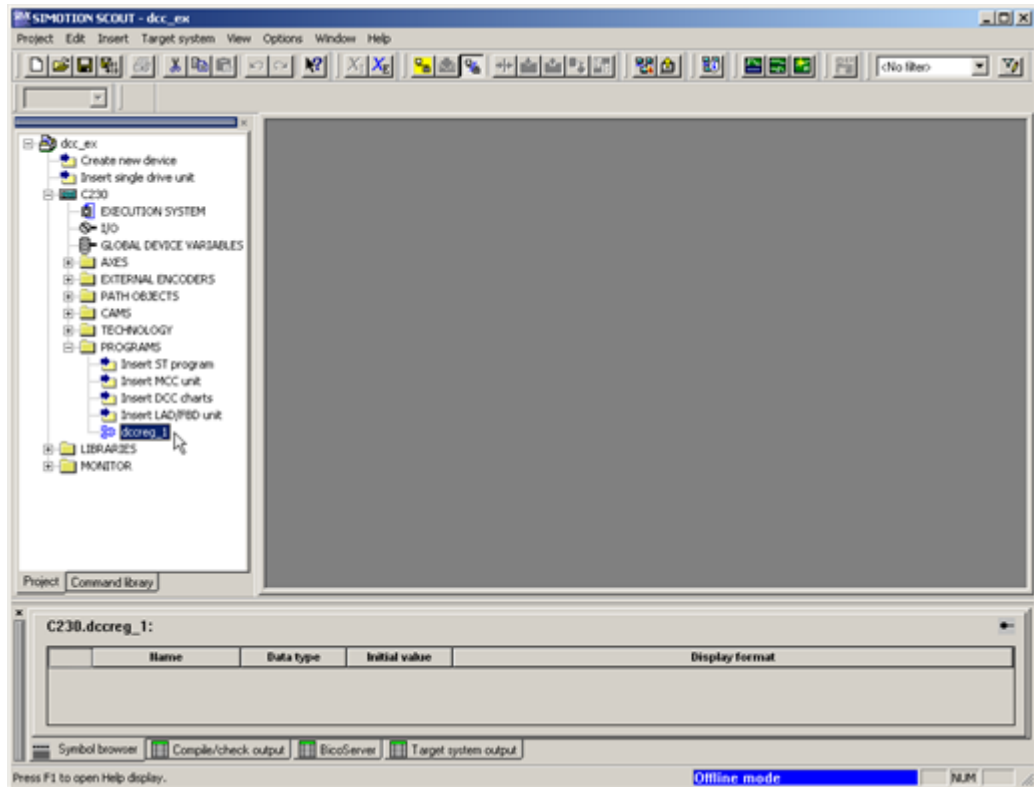
Figure 5-6   Inserting a chart

This creates the project structure and a chart. All that remains is to create some activity within the chart, i.e. by inserting blocks and interconnecting them.

5.2.4 Inserting blocks
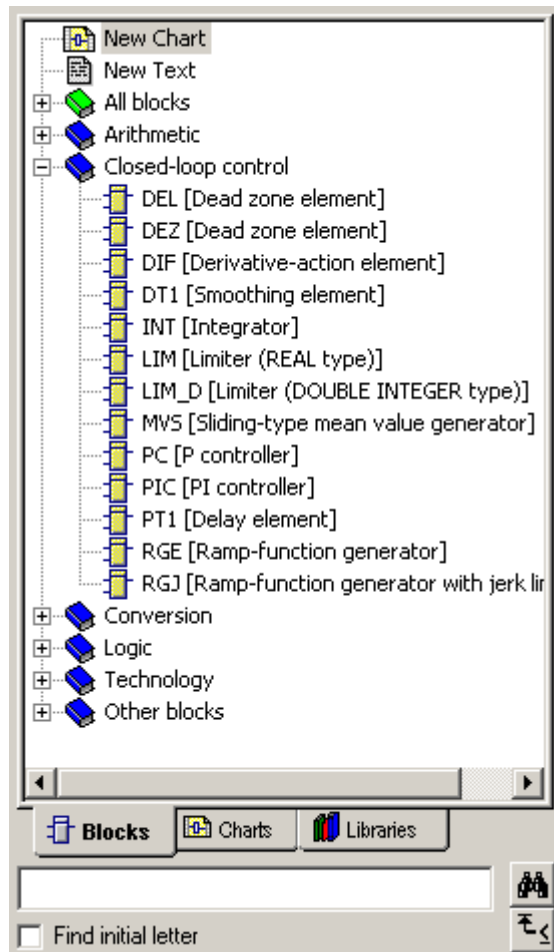
Figure 5-7   Inserting a block

- Open a block family in the **Closed-loop control** family with the closed-loop control blocks.

- Select the required block and insert it in the chart using drag-and-drop. Only the outline of the block in dashed lines is displayed during the copying procedure. Release the mouse button at the required point.

- To search for a block, enter its name in the input field of the block catalog and click the **binoculars** button. The search process begins. Once the block has been found, insert it in an empty space on the chart using drag-and-drop.

Note

If blocks are superimposed on the chart with other elements, such as other blocks or the sheet bar, the superimposed block will be displayed in gray and its connections will not be visible. You must reposition the blocks to ensure that all block information can be viewed.

Changing to the page view

To change to the page view from the overview representation, right-click an empty space in the chart and select **Display this page** in the context menu that appears. The names of the block connections are displayed in this enlarged view.

You can also switch to the page view and back to the overview again by double-clicking an empty area on a page.

Configuring the block display

You can change the display of the blocks. You can change the block width using **Options > Settings > Blocks / sheet bars width....** You can change the designation of the inputs and outputs of the blocks using **Options > Settings > Display** in the **Connections** submenu.

The block type can be displayed in the form of both text (name) and graphics (control symbol). This can be configured via **Options > Settings > Display** in the submenu **Block headers**.

5.2.5   Interconnecting blocks

Procedure

- Select the **Y** output of the first **INT**egrator, followed by the **X** input of the second **INT**egrator.
- Select the **Y** output of the second **INT**egrator, followed by the **X** input of the inverter (**SII**).
- Select the **Y** output of the inverter (**SII**), followed by the **X** input of the first **INT**egrator.

The autorouter creates the connecting lines from the outputs to the inputs and they are then interconnected.

5.2.6   Parameterizing block connections in the chart

For the first integrator INT 1/1, the following initial values should be assigned to the connections: LL = -10.0, LU = 10.0, SV = 2.0, Ti = 100 ms

For the second integrator INT 1/2, the following initial values should be assigned to the connections: LL = -10.0, LU = 10.0, Ti = 100 ms

To do this, open the **Properties - Connection** window of the associated block connection with a double-click. Enter the initial value for **Value** and click **OK** to close the window. Note that for the input of the values for Ti, the "ms" unit must follow the numeric value 100 without any spaces.

As the connections mentioned above are not interconnected, the entered values also remain valid during the cyclic processing of the chart.
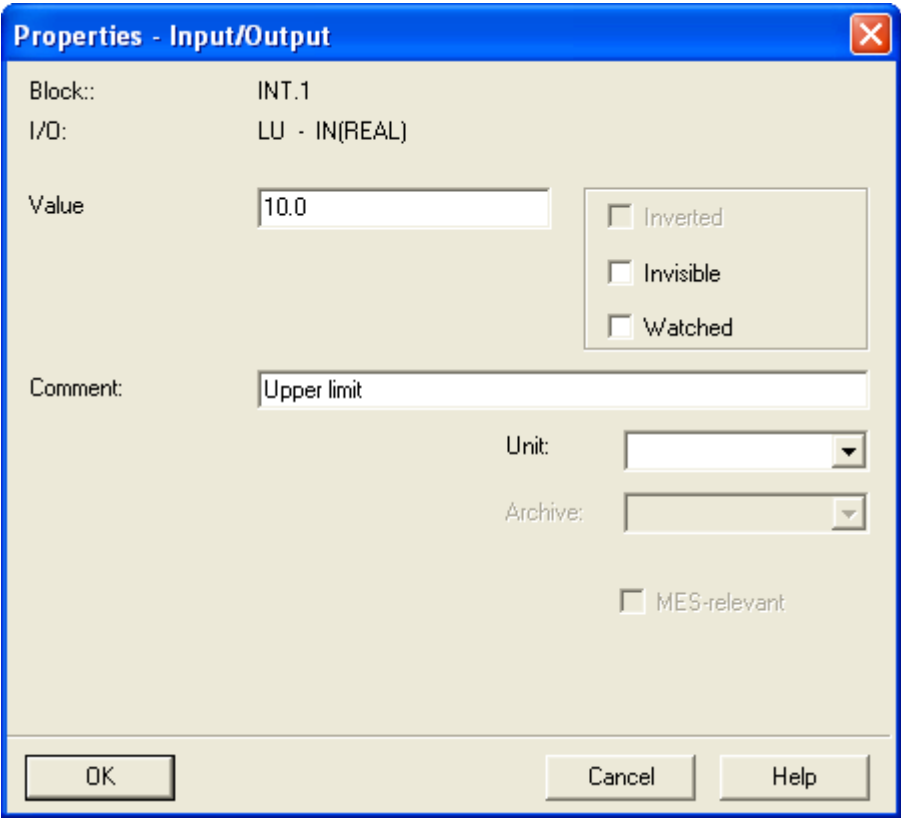
Figure 5-8   Properties window of the "LU" block connection
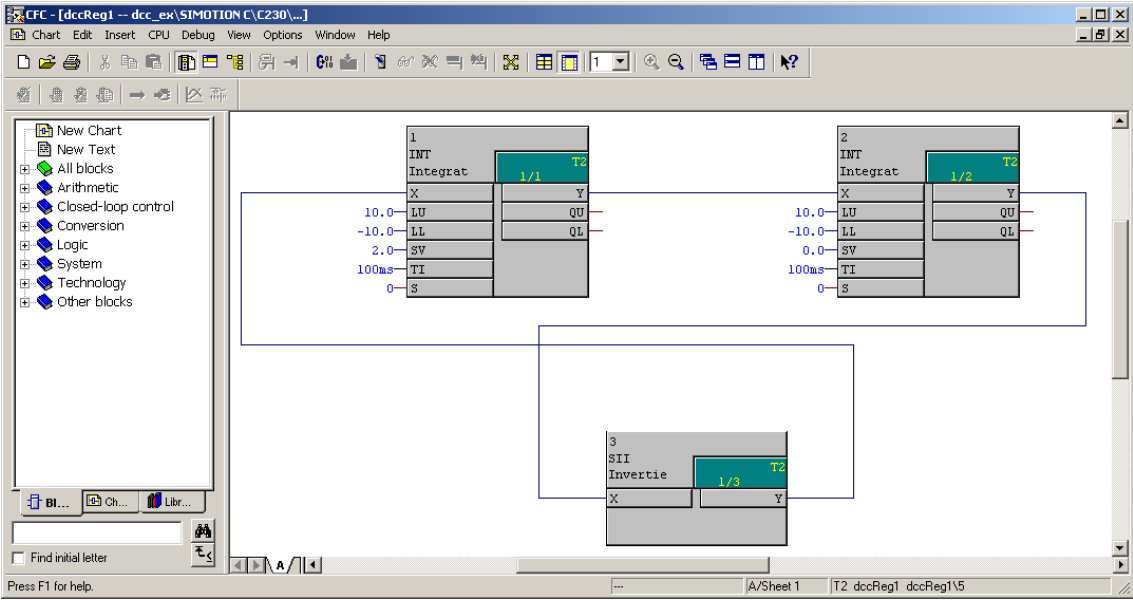
Figure 5-9   Chart "dccReg1" with interconnected blocks. Some connections have been assigned values that differ from the default values.

5.2.7 Structures for DCB block connections

As of DCC 2.1/SIMOTION RT 4.2, user-defined data types structured for DCC SIMOTION can be used for block connections. The structure definitions are taken from the DCC block library.
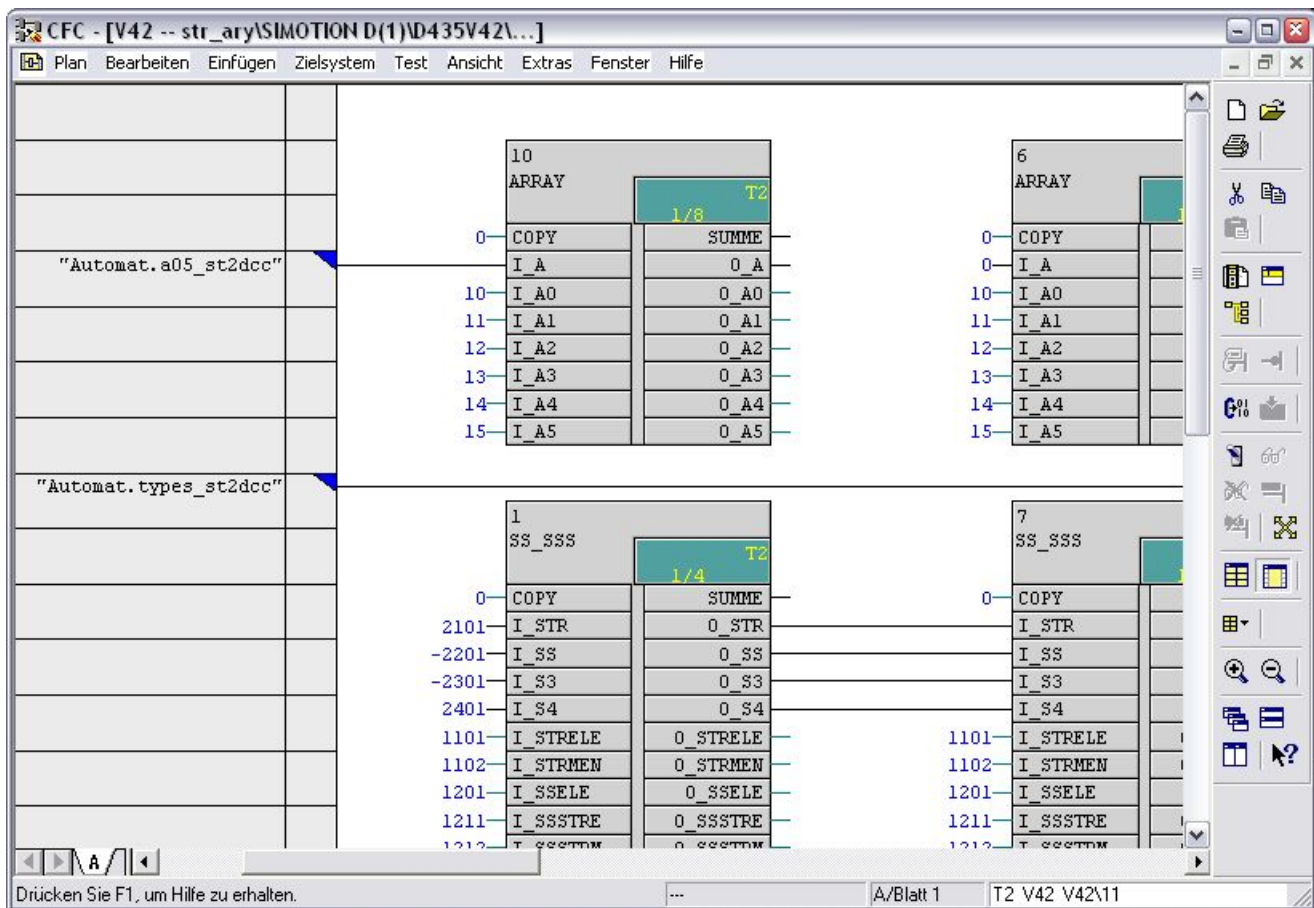
Figure 5-10   Interconnected structures

Purpose

Signals associated with one another from a technological perspective can be combined to form structures.

For example, this means that a setpoint, factor, and control signal can be transferred as related components of the same structure. The structure can be extended at block level; this means there is no need to make any changes in the DCC chart.

Creating structures

New structures can only be generated by blocks that have been created with the SIMOTION CLib Studio. Structures cannot be defined in the DCC editor.

Transferring structured block connections to/from TO variables

The structures must be the same in the DCB library and in the TP definition.

It is not possible to accept structures from TPs. The structure must be defined in the DCB block.

Assigning parameters for structured block inputs

An initialization value is pre-assigned to structured block inputs.

Call up the object properties for a structure or an elementary data type in the structure in order to edit them. Double-click the block connection to open the **Select structure element** dialog. Highlight the structure element you wish to edit and open the **Properties** dialog by either clicking the **Properties** button or double-clicking the structure element.

The **Properties** dialog can be called up in both edit mode and test mode.
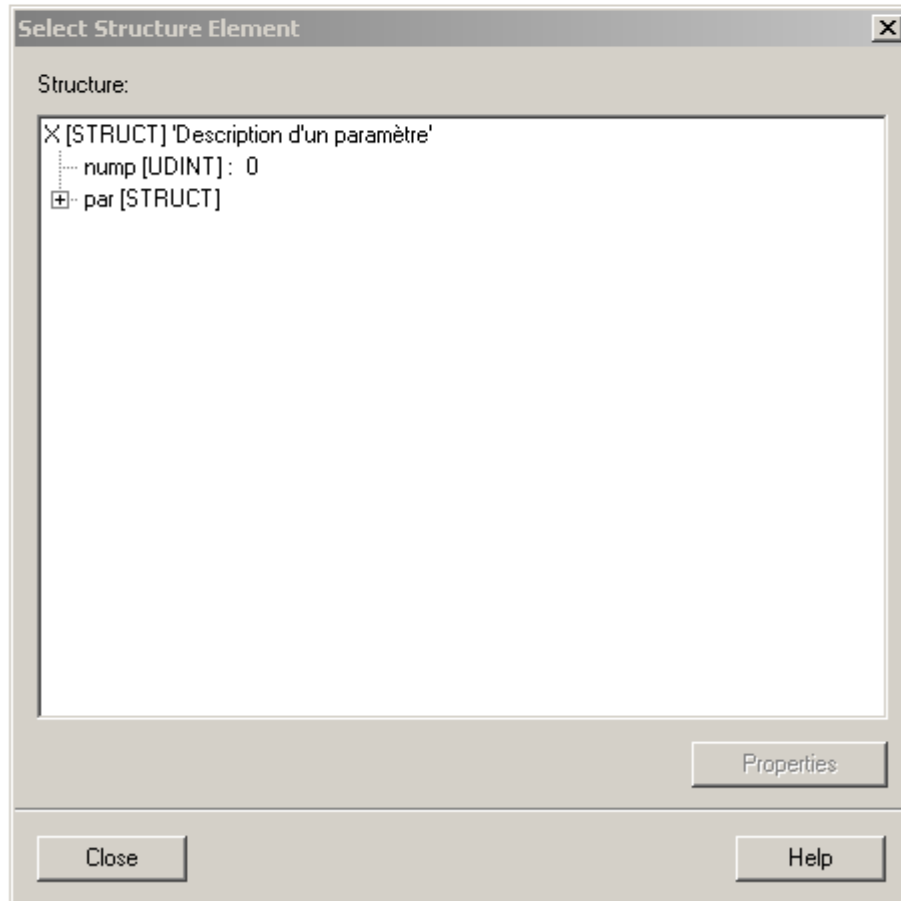
Figure 5-11   Assigning parameters for structured block connections

Arrays are represented by individual symbols for display in the CFC editor.

Example: FIELD|2|.
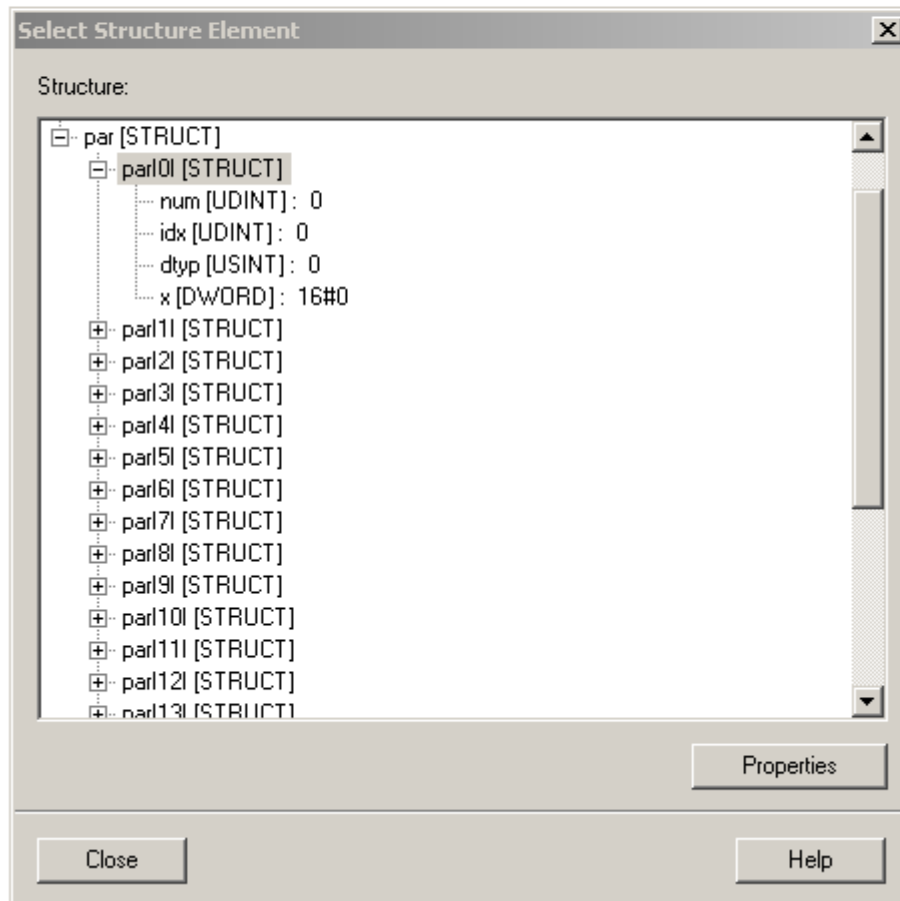
Figure 5-12   Assigning parameters for structured block connections

Double-clicking a structure element or clicking the **Properties** button opens a dialog box that is similar to the one for a block connection of an elementary data type.

Interconnecting structured block connections

Structured block connections are interconnected with the following in the editor:

● Connections of the same type from other CFC blocks

● Global variables of the same type

- TO variables of the same type

- IO arrays of the same length

Structured connections can only be interconnected as an entire unit.

It is not possible to interconnect individual structure elements.

You can monitor values using the dynamic display.

Structures can be interconnected when the name of the data type and the basic data type of the individual elements match. The interconnection rules of the DCC apply: DINT and DWORD are considered to be the same type.

Arrays can be interconnected when the basic data type and the dimension match.

With arrays of structures, the complete definition of the structure must be identical so that the connections can be interconnected.

Structures can only be monitored in the values table.

Interconnection of structured block connections to external variables

You can also interconnect structured block connections to external variables, e.g. a global variable of an ST block. However, as only structures from a DCBLib are valid, this library must be imported within the ST source file and the structure type from the DCBLib used for the global variable. The use of an identical reproduction within ST is not possible for an interconnection.

Example of an ST source file for the interconnection of input X of the RMDP DCC block

```
INTERFACE
    USEPACKAGE dcblib_simotion_4_3;
    VAR_GLOBAL
        x: DCB_RMDP_PAR;
    END_VAR
END_INTERFACE
```

An input or output parameter of a block (in this case RMDP) with the appropriate structure type can then be interconnected to the global ST variable via **Interconnection to operand....** For subcharts, the input of the block must already have been interconnected previously to an input or output parameter of the subchart.

Publishing structured block connections

You can publish a structure element by entering a comment beginning with an @ sign at the structure element.

You can publish a block connection by entering a comment beginning with an @ sign at the block connection (top level). A structured variable is then created. The first identifier in the comment determines the variable identifier.

Note

DCB blocks with user-defined structures may only be edited as of SIMOTION 4.2.

Libraries with structures may not be generated for older SIMOTION versions.

Block types with user-defined structures may not be used in DCC SINAMICS.

5.2.8   Default connection values for delta downloads

Changes made to the default values of the block connections at a later date are only transferred to SIMOTION if the option highlighted in the following image is activated:
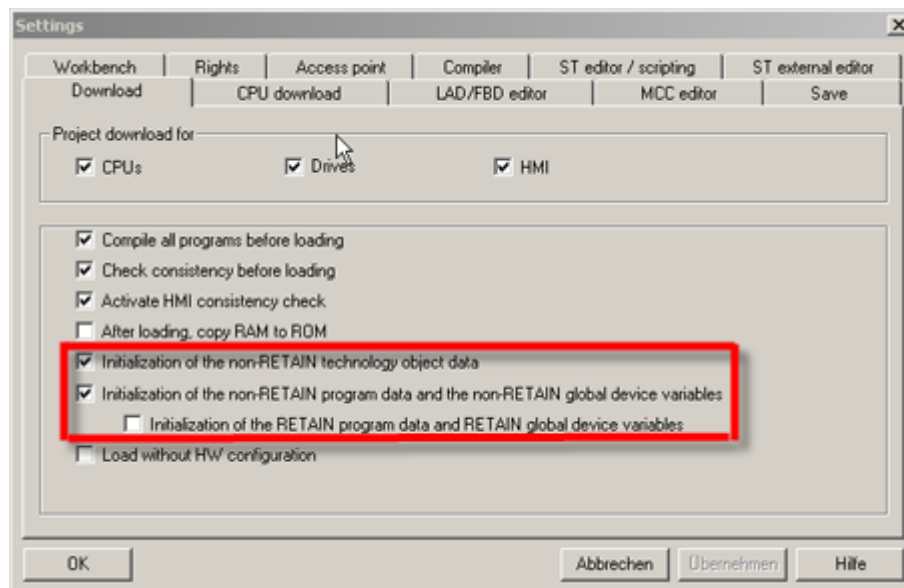
Figure 5-13   Settings for SIMOTION data transfer

5.2.9 Compiling the DCC in the DCC editor
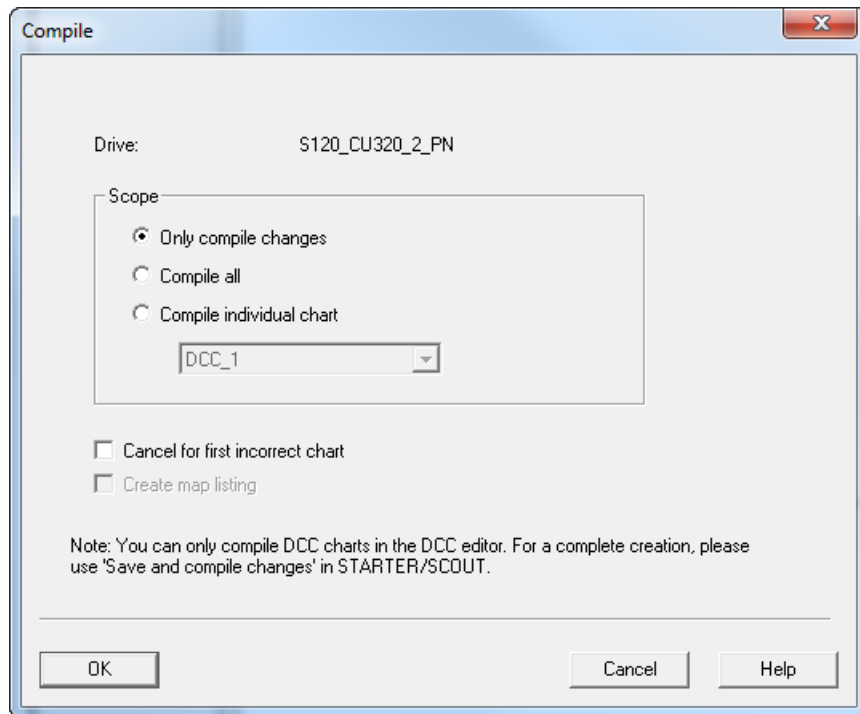
Compiling

Figure 5-14   Compiling a dialog in the DCC editor

You can begin compiling with **Chart > Compile > Charts as program...** or via the [icon] button.

If errors occur during compilation, the **Logs** dialog box will automatically be displayed at the end of the procedure (just as in the case of the consistency check).

Note

You can check whether all the required block libraries and technology packages have been activated via the menu item **Options -> Block types**.
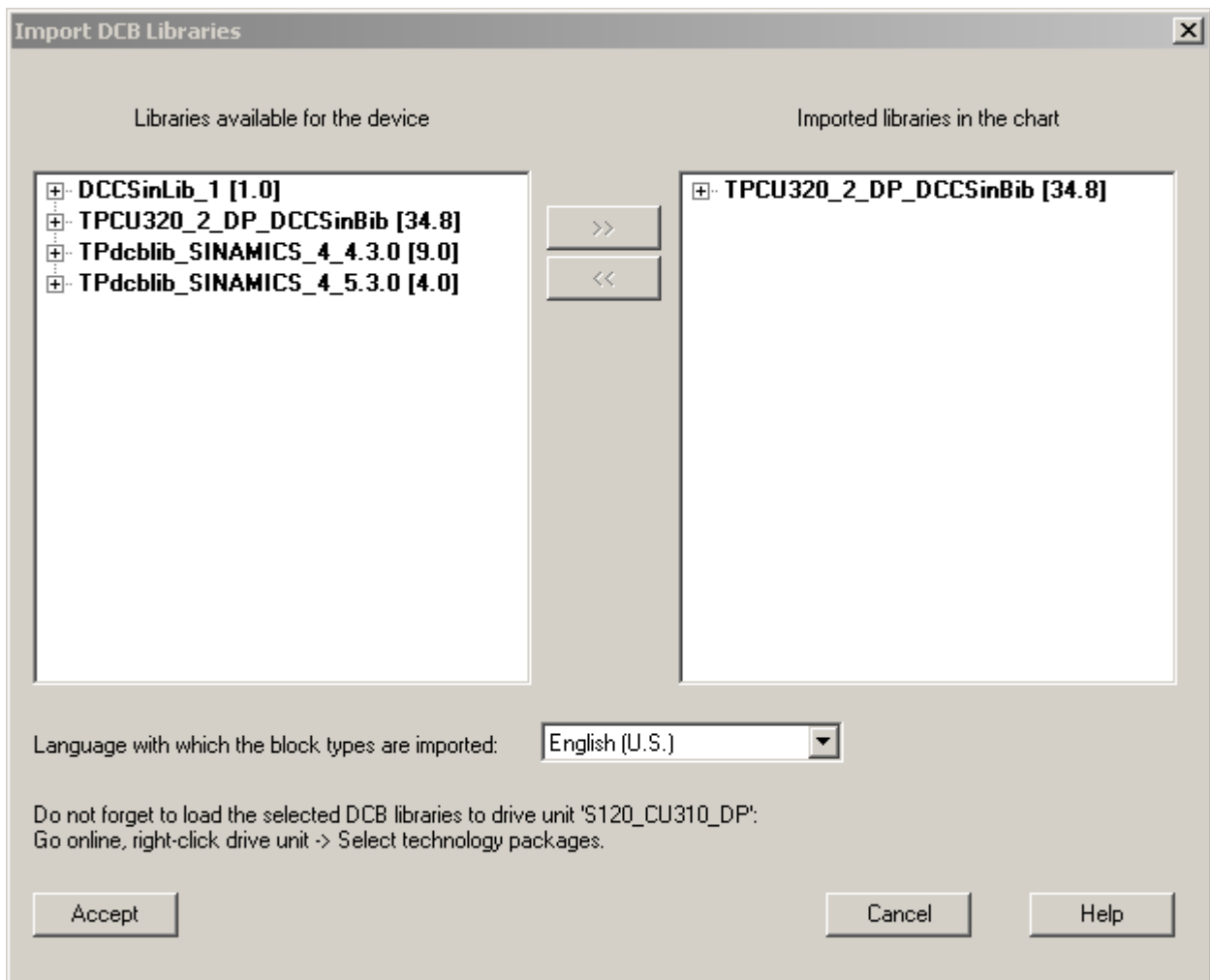
Figure 5-15   Updating the block library dialog

Compilation options

For detailed information about the compilation options, refer to Compiling (Page 51).

> **Note**
>
> The **Reorganize HMI variable interface** option enables you to reassign the addresses for all @ variables currently defined in the chart. @ variables that are still available in the interface, but are no longer used in the charts, are deleted.

After compilation

On completion of the compilation procedure, a detailed compilation log appears.

● To access the relevant block that is causing an error, select the error line in the log and click **Go to**.

● The log can be called again at a later point via the menu **Options > Logs** and, if necessary, can also be printed out.

5.2.10 Loading the compiled DCC

To be able to operate the DCC program on a CPU, it must have first been downloaded to the CPU.

Downloading can only take place in SIMOTION SCOUT.

Note

The DCC editor assigns the configured DCBs to tasks T1 to T5 in the execution system.

5.2.11 Starting the CPU

To execute the DCC program, the CPU must be switched to RUN.

5.2.12 Enable attribute, execution groups

The "Enable" attribute enables or disables an execution group (enable = 1; disable = 0).

The "Enable" attribute is set to 1 by default. However, it can also be set dynamically. The output value of a block determines whether the group is to be enabled or disabled. To this end, you can interconnect the digital output of a block with the execution group.

5.2.13 Creating documentation

Complete documentation

The example has been successfully configured and tested. You can now create the complete documentation for your example.

Chart reference data

In this case, the chart reference data is the **block type cross references** and the **execution sequence**. In the toolbar, click:

You can open the **Chart Ref: Display Chart Reference Data** window via **Options > Reference data** in the menu. In this window, you can use the buttons in the toolbar to create and display the relevant lists. To begin printing, click: 🖨

Then close the dialog box with **Reference data > Exit**.

Note

The **Chart Ref: Display Chart Reference Data** window can only be opened when the window of the execution editor is not open.

Printing a chart

You also want to print the chart. Since only one page has been used on this chart, a single page display is recommended for printing. The chart is now set to **Page view**. To begin printing, click: 🖨

If your chart consists of several pages, we recommend that you print them individually in the page view. Click the 🖼 button to change to the page view.

Click 🔍 or 🔍 to customize the display so that all DCBs are shown on a single page on the PC monitor.

See also

Compiling the DCC in the DCC editor (Page 224)

5.3 DCC SIMOTION specifications

5.3.1 Rules for assigning names in the DCC editor

Variables are used for data exchange between SCOUT and the DCC editor. The variable names used are subject to the same restrictions regarding names in SIMOTION.

Note

In the DCC editor, the length of the name is restricted to **a maximum of 49 characters**. In SIMOTION, however, longer names are supported. Where this arises, it is necessary to either use aliases or interpose ST variables.

The names in the DCC editor must therefore abide by the following rules:

- Basic chart:
  - No keyword or previously defined name permitted
  - Must start with a letter
  - May contain numbers, letters, and underscores
  - "_" must be followed by a number
- Subchart:
  - Must start with a letter
  - May contain numbers and letters
  - Underscores are not permitted
  - Keyword or previously defined name permitted (during compilation, a check is performed to determine whether this is unique)
  - During compilation in the DCC editor, the name is checked to determine whether it is unique
- Block instance:
  - Can start with a number
  - May contain numbers and letters
  - May not start with an underscore
  - Keyword or previously defined name permitted
  - In the DCC editor, a check is performed to determine whether the name has previously been defined

- Execution group:
  - Keyword or previously defined name permitted
  - In the DCC editor, a check is performed to determine whether the name has previously been defined
- Chart connection:
  - No keyword or previously defined name permitted
  - In the DCC editor, a check is performed to determine whether the name has previously been defined

5.3.2   Field/name lengths and conventions

Field/name lengths and conventions

Object	Length	Remark
Chart name	22 *)	May not contain the following characters: \ . : / * ? " < > | # % ( ) Use of the "_" character is subject to particular specifications.
Chart comment	255	All ANSI characters are permitted.
Execution group	22	Only letters, numbers and underscores may be used.
Block type	6	Determined by DCC.
Block instance comment	80	All ANSI characters are permitted.
Parameter comment	80	All ANSI characters are permitted.
Block name (instance name)	16 *)	May not contain the following characters: \ . : / * ? " < > | # % Use of the "_" character is subject to particular specifications.
Global operand	Max. 49	If names from SIMOTION are too long, an ALIAS must be defined.

*) The chart name and block name must jointly consist of a maximum of 24 characters, including separating characters.

See also

Rules for assigning names in the DCC editor (Page 228)

5.3.3 Representation of the dynamic value display

The values are output next to the connections, according to their data type. They are displayed on the screen with a colored background.

Table 5-2   Representation of the dynamic value display

Representation	Meaning
Blue on white	Representation of the values in edit mode (offline)
Black asterisks on yellow	Values during transfer to the dynamic display
Black value on yellow	Representation of the values read from the SIMOTION RT in test mode
#### on a red background	Representation of values while the dynamized values required from the SIMOTION RT are missing (fault, overload)

5.3.4 Displaying operands

When integrating OpenCFC into SIMOTION DCC, only symbolic operand names rather than absolute operand names may be used. The settings for displaying operands in the CFC Editor under **Options -> Settings -> Display ...** are ignored.

5.4 Faults and warnings

5.4.1 Notes on error message display

Notes on error message display

If an error message or warning appears in the DCC, there will be special characters that draw your attention to names or parts of names. You can double-click these characters to proceed directly to the relevant site of the error.

The characters have the following meanings:

- Block instance or block input/block output

    – Represented as:
    "\." <instance name> ["." <pin name>]

    – Example:
    Interconnection of "_device\ST_1.i" with "._DCC_6_CFC1_1.x_1" cannot be achieved; invalid type

- Interconnection to chart connections

    – Represented as:
    "<" <variable name> ">"

    – Example:
    The input connection DCC_5\<interconn_add_1_x_1> from the chart 'DCC_5' is not interconnected with a sink.

- Execution level/execution group

    – Represented as:
    "{" <group name> "}"

    – Example:
    Empty execution group {EG}

Note

Error analysis when a DCB block crashes

If a DCB block crashes during operation, information about the cause and the block instance can be found in the device diagnostics buffer in SIMOTION. With this information, you can contact the creator of the block library, remove the relevant block instance or change the parameterization.

Appendix

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

A.1 List of abbreviations

Abbreviation	Description
(G)UI	(Grapical) User Interface
Connection	General term for block input or block output
BI parameter	Binector Input parameter. The parameter is used for the interconnection of a binector to a sink signal that can only have the states 0 or 1
BICO	Binector-Connector that designates an interconnectable parameter in the drive
BO parameter	Binector parameter (also Binector Output parameter). The parameter can be used as a binary signal source (0 or 1)
CFC	Drive Control Chart
CI parameter	Connector Input parameter. The parameter is used for the interconnection of a connector to a sink signal
CO parameter	Connector parameter (also Connector Output parameter). The parameter can be used as a signal source
CSV	Comma Separated Value, text format for column-oriented data
DCB	Drive Control Block
DCC	Drive Control Chart
DO	Drive Object
ELF file	File coded in Executable Linkable Format
FEAT	Feature document
ITCP	Instance, Time slice, Connection and Parameter/Variable
MBCS	Multi-Byte Character Set, corresponds to UTF-8
MDI	Multiple-Document Interface – an application with several windows
OEM	Original Equipment Manufacturer
TP	Technology Package

A.2 Glossary

Execution level	Interface to the execution system used for the execution of blocks.
Execution group	Execution groups are used for structuring or subdividing execution levels. The blocks and/or charts are integrated sequentially into the execution groups. SIMOTION only: Execution groups, for example, can be switched on and off separately through a connection to a block output (data type BOOL). When an execution group is switched off, all blocks/charts contained therein are no longer calculated.
Execution sequence	Sequence in which the blocks should be calculated in an execution level or execution group
User library	A user library which contains block libraries. As the blocks are based on charts, they can be handled individually in this case.
Output connections	The output connections can be interconnected to other inputs or assigned an initialization value. The value is then active at this connection when the block is calculated in the INIT mode for the first time. This is useful, for example, when a specific value is to be assigned to the output connection of a flip-flop block or a controller block.
HMI variables	You can declare block inputs and outputs as **HMI variables** and therefore generate a static interface for these for use in your system visualization.
Basic library	A library that has been created in C, e.g. with the aid of a DCB tool. Such a library is a closed unit, i.e. a DCB cannot, for example, be separated from a library in order to transfer it to another block. The supplied DCBLIB is an example.
Basic chart	The chart that is visible and can be managed in STEP 7 or SCOUT/STARTER. All other charts, i.e. chart partitions or subcharts can only be managed in the DCC editor. Only the term chart is used in the following.
Block help	Detailed information about a certain block type can be obtained by selecting the block and pressing the F1 key.
Block type library	Consists of the description of the block types contained in the library and the object file that implements the blocks, e.g. ELF.
INIT mode	Initialization, ramp-up
RUN mode	Cyclic operation
DCB library	A user-defined DCB is created from a chart.
DCB studio	Development environment for the programming of C block libraries.
DCB viewer	You can read the descriptions of the individual block types in the DCB viewer during the configuration in the DCC editor. The DCB viewer is opened by selecting a block and pressing the F1 key.

Input connections	The input connections can be parameterized with constants by the project engineer or interconnected to other block outputs. When the DCB blocks are called, the inputs and outputs have already been assigned default values that can be changed when required.
Creation mode	DCCs can be edited in creation mode.
Freely defined parameters	@ parameters and @ technology connectors. Interface defined for the parameterization by the user in the form of parameter numbers that can be defined as ALIAS for block inputs and block outputs.
Global operands	Global operands are connection partners for block inputs/outputs with which DCB blocks can read and write information from the system environment.
Hierarchical chart or also subchart	A chart (subchart) can be inserted in another chart (chart-in-chart-technique). Hierarchical structures can be formed here. Each inserted chart can be opened and, like every other chart, edited further and therefore changed individually. A chart can be encapsulated for further use, i.e. chart connections added. Which block connections are provided at the chart connections can also be specified individually.
Comments	Each block connection of the CFC page can have a comment added.
Connector	If no more lines can be drawn because the page is too full, the CFC inserts a connector at the block/chart connection and a number in the sheet bar. The corresponding connectors are assigned the same reference number. If several interconnections that cannot be displayed all start at one output, they are all assigned the same reference number. You can recognize the connection point location through the varying representation of the connector. For details, refer to the CFC help.
Laboratory mode	The blocks are automatically monitored in laboratory mode (display of the current values of the block/chart connections logged on for testing).
Offline	Project is processed without a connection to the device.
Online	Project is processed with a connection to the device.
Online RUN, or pulse enable at the drive	Project is processed with a connection to the device and the device is in open-loop or closed-loop control.
Parameterization	Instead of an interconnection, a constant different than the default can be parameterized at each input or output.
Process mode	The blocks are not automatically monitored in process mode (display of the current values of the block/chart connections logged on for testing) so that the additional load is very small.
Sheet bars	The sheet bars on the left and right sides of a CFC page contain the references to the interconnected objects, e.g. other blocks or execution groups, that are not on the current page. They also contain the number of the connector (breakpoint) when the connection line to the sheet bar cannot be drawn because the page is too full.
Technology package	A technology package (the DCBLIB) contains technology objects (DCB) that are capable of being instantiated.

Chart partition	Each chart comprises up to 26 chart partitions ("A".."Z"). Each chart partition consists of up to six DIN A4 pages. A newly created chart only contains chart partition "A".
Test mode	Test mode can be used to debug the DCCs.
Text interconnections	Are used to split projects into separate, configurable units and define an "open" interconnection between charts.
	A text interconnection can only be at a block/chart input and always references a block or chart output in CFC. The text interconnection is an "open" interconnection until it becomes a "real" interconnection upon closing.
	A closable text interconnection is the addressing of an input by means of a character string, which identifies a specific interconnection source (output) via the path specification (chart/block.connection).
Trace	The signals of the block outputs can be recorded with the trace function.
Trend display	In test mode with SIMOTION, you can use the value and trend display to analyze the input and output values of blocks.
Typical	DCB library
Overflow pages	Overflow pages are created automatically when more sheet bar entries are generated than can be displayed on a page. An overflow page consists solely of the sheet bars and contains no further objects.
Interconnecting	The term interconnecting means:
	The connection of a block output to another DCB block input on the same device
	Interconnections of a block output to an execution group. SIMOTION only
	Interconnection of a block output to a global operand or a global operand to a DCB block input. A global operand can be the following:
	The name of a SIMOTION variable.
	The name of a SINAMICS parameter (can also be performed implicitly through the interconnection to a SB).
Workbench	The workbench is the navigation center for the individual engineering steps and is used for the centralized creation and management of projects. It provides a uniform, function-oriented, consistent, filterable view of all data and programs, even in distributed systems.

Index