SIEMENS

SINUMERIK

SINUMERIK ONE
Create MyConfig - Diff, Expert, Topo

Operating Manual

Introduction 1

Security instructions 2

Create MyESW 3

Installation 4

Uninstallation 5

Create MyConfig - CMC script 6

Create MyConfig - Diff 7

Create MyConfig - Expert 8

Create MyConfig - packages 9

Create MyConfig - Topo 10

FAQs 11

List of abbreviations A

Valid for:
CMC Software Version 6.3
SINUMERIK ONE Version 6.15 SP1

01/2022
A5E50115644B-AC

Legal information

Warning notice system

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

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

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

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

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

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

Qualified Personnel

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

Proper use of Siemens products

Note the following:

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

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E50115644B-AC
Ⓟ 01/2022 Subject to change

Copyright © Siemens AG 2019 - 2022.
All rights reserved

Table of contents

1 Introduction 11

1.1 About SINUMERIK 11

1.2 About this documentation 11

1.3 Documentation on the internet 13

1.4 Feedback on the technical documentation 13

1.5 mySupport documentation 13

1.6 Service and Support 14

1.7 Important product information 16

2 Security instructions 17

2.1 Fundamental security instructions 17
2.1.1 General security instructions 17
2.1.2 Warranty and liability for application examples 17
2.1.3 Security information 17

2.2 Product-specific security instructions 19
2.2.1 Password handling for CMC V6.3 not supported 19
2.2.2 Administer the computer according to general security guidelines 20
2.2.3 CMC Shield 20

3 Create MyESW 21

4 Installation 23

4.1 Overview 23

4.2 Hardware requirements 23

4.3 Software requirements 23

4.4 User-guided setup 24

4.5 Performing the automated installation 24

4.6 Extensions to CMC 24

4.7 Starting Create MyConfig 25

5 Uninstallation 27

6 Create MyConfig - CMC script 29

6.1 Introduction 29

6.2 Script Editor 29

6.3 #include - script 30

6.4 Locations where the script languages is used 31

6.5 Color coding of language elements 32

6.6 System variables - CMC scripts in Diff ... 34
6.6.1 Overview of system variables ... 34
6.6.2 Using system variables ... 34
6.6.2.1 Up.$Env.BatchMode ... 35
6.6.2.2 Up.$Script.ScriptFile ... 35
6.6.2.3 Up.$Script.ArcFile ... 35
6.6.2.4 Up.$Script.Result ... 36
6.6.2.5 Up.$Script.EditMode ... 38
6.7 Script libraries ... 38
6.8 CMC standard script libraries ... 39
6.8.1 CMC standard script libraries ... 39
6.8.2 SINAMICS library ... 40
6.8.3 Utilities library ... 40
6.9 Example exporting archive data ... 40
6.10 Example, manipulating archive data ... 42
6.11 Package variables ... 44
6.11.1 Script variables ... 45
6.11.2 DO variables ... 46
6.11.3 System variables ... 47
6.12 Functions and procedures ... 47
6.13 Commenting ... 49
6.14 Area data ... 49
6.15 Data accesses ... 50
6.15.1 Sections ... 53
6.16 Control structures ... 54
6.16.1 Control structures - context menu ... 54
6.16.2 If control structure / If-Else control structure ... 55
6.16.3 If-ElsIf control structure / If-ElsIf-Else control structure ... 55
6.16.4 While control structure ... 56
6.16.5 Pragma Eol control structure ... 57
6.16.6 Rem control structure ... 57
6.17 Operators ... 57
6.17.1 Overview operators ... 57
6.17.2 Assignment operators ... 58
6.17.3 Arithmetic operators ... 59
6.17.4 Logic operators / comparison operators ... 60
6.17.5 Special operators ... 61
6.17.6 String concatenation ... 62
6.18 String operations ... 62
6.18.1 StringLen ... 62
6.18.2 StringMatch ... 63
6.18.3 StringPos ... 63
6.18.4 StringReplace ... 64
6.18.5 StringSubStr ... 65
6.19 File operations ... 65

6.19.1 FileCopy ... 66
6.19.2 FileDelete ... 67
6.19.3 FileExist ... 67
6.19.4 FileRead ... 69
6.19.5 FileWrite ... 70
6.19.6 QueryIni ... 72
6.19.7 QueryXml ... 74
6.19.8 TraceToFile ... 76

6.20 Dialogs ... 77
6.20.1 Msg ... 78
6.20.2 Warning ... 79
6.20.3 Error ... 79
6.20.4 Introduction input functions ... 80
6.20.5 Input ... 80
6.20.6 InputChoice ... 81
6.20.7 InputEnum ... 84
6.20.8 InputInt ... 85
6.20.9 InputReal ... 86
6.20.10 InputText ... 88
6.20.11 InputUInt ... 89
6.20.12 Text formatting options ... 90

6.21 Miscellaneous ... 97
6.21.1 Return ... 97
6.21.2 ExtCall ... 97
6.21.3 Redo ... 98
6.21.4 Skip ... 98
6.21.5 DateTime ... 99
6.21.6 DOVar ... 100
6.21.7 Log ... 101
6.21.8 Logging ... 101
6.21.9 MathRound ... 101
6.21.10 Version ... 102

6.22 From library ... 103
6.22.1 SINAMICS ... 103
6.22.1.1 SINAMICS library ... 103
6.22.1.2 SinamicsTecActivate ... 103
6.22.1.3 SinamicsTecDeactivate ... 104
6.22.2 Utilities ... 104
6.22.2.1 Utilities library ... 104
6.22.2.2 IsBit ... 104
6.22.2.3 SetBit ... 105
6.22.2.4 ClrBit ... 105

7 Create MyConfig - Diff ... 107

7.1 Introduction ... 107

7.2 SINUMERIK archive comparison ... 108
7.2.1 General information on the SINUMERIK archive comparison ... 108
7.2.2 Overview of user interface ... 109
7.2.2.1 Main menus ... 109
7.2.2.2 SINUMERIK archive comparison - Toolbar ... 117

7.2.2.3 SINUMERIK archive comparison - Overview window ... 120
7.2.3 Configuration ... 127
7.2.4 Copying/exporting data from the SINUMERIK archive comparison ... 134
7.2.5 Pasting/importing data in the SINUMERIK archive comparison ... 137
7.2.6 Data tasks in the SINUMERIK archive comparison ... 139
7.2.6.1 Overview ... 139
7.2.6.2 Configuring data jobs ... 140
7.2.6.3 Executing data tasks ... 144
7.2.7 Data class archives ... 148
7.2.8 CMC scripts in Diff ... 151
7.2.8.1 Introduction ... 151
7.2.8.2 Data accesses / sections / identifiers ... 151
7.2.8.3 System variables ... 152
7.2.8.4 Calling scripts from the interface ... 152
7.2.8.5 Calling scripts from command line ... 153

7.3 Folder comparison ... 154
7.3.1 General information ... 154
7.3.2 Overview of user interface ... 154
7.3.2.1 Main menus ... 154
7.3.2.2 Folder comparison - toolbar ... 162
7.3.2.3 Folder comparison - overview window ... 165
7.3.3 Changing the column layout ... 168
7.3.4 Configuration ... 169
7.3.5 Copying from the folder comparison ... 173
7.3.6 Save archive as user or SDB archive ... 174

7.4 File comparison ... 175
7.4.1 General information ... 175
7.4.2 Overview of user interface ... 175
7.4.2.1 Main menus ... 175
7.4.2.2 File comparison - toolbar ... 184
7.4.2.3 File comparison - overview window ... 187
7.4.3 Configuration ... 189
7.4.4 Configuration of the automatic file identification ... 195
7.4.5 Copying from the file comparison ... 197
7.4.6 Display modes ... 198

7.5 Exporting and importing configurations ... 198
7.5.1 General information ... 198
7.5.2 Exporting ... 199
7.5.3 Import ... 199

7.6 Command line call ... 200
7.6.1 Diff ... 200

7.7 Using regular expressions in Diff ... 206
7.7.1 General information on regular expressions ... 206
7.7.2 Meta characters in the regular expressions ... 207
7.7.3 Examples of regular expressions ... 209

8 Create MyConfig - Expert ... 213

8.1 Introduction ... 213
8.1.1 Overview of the Expert user interface ... 214
8.1.2 Easy to operate Expert user interface ... 223

8.1.3 Check run ... 229
8.1.4 Opening a project from CMC Version 6.0 ... 230

8.2 Creating/configuring a project ... 230
8.2.1 Opening Expert ... 230
8.2.2 Creating a new project ... 232
8.2.3 Project explorer ... 234
8.2.4 Specifying the configuration for Package ... 236

8.3 Dialogs ... 239
8.3.1 Overview ... 239
8.3.2 Dependency of the dialog pages on data areas ... 240
8.3.3 Dialog scripts ... 242
8.3.3.1 Introduction ... 242
8.3.3.2 Meaning of the events ... 245
8.3.3.3 Executing scripts in the dialog pages ... 245
8.3.3.4 OnInit event ... 247
8.3.3.5 OnNext event ... 248
8.3.3.6 OnEnd event ... 250
8.3.3.7 Editing scripts ... 251
8.3.4 Package configuration (configuration) ... 253
8.3.5 Package configuration (package execution) ... 254
8.3.6 Notes on the package (configuration) ... 256
8.3.7 Notes on the package (package execution) ... 257
8.3.8 Access to target system (configuration) ... 259
8.3.9 Access to target system (package execution) ... 263
8.3.10 Select offline archive (configuration) ... 266
8.3.11 Select offline archive (package execution) ... 272
8.3.12 Backup initial state (configuration) ... 273
8.3.13 Backup initial state (package execution) ... 276
8.3.14 PCU setup (configuring) ... 277
8.3.15 PCU setup (package execution) ... 278
8.3.16 CNC software (configuration) ... 279
8.3.17 CNC software (package execution) ... 283
8.3.18 System configuration (configuration) ... 285
8.3.19 System configuration (package execution) ... 297
8.3.20 SINAMICS topology (configuration) ... 299
8.3.21 SINAMICS topology (package execution) ... 304
8.3.22 Version Display (configuration) ... 309
8.3.23 Version Display (package execution) ... 311
8.3.24 Step tree (configuration) ... 312
8.3.25 Step tree (package execution) ... 314
8.3.26 Archive data transfer (package execution) ... 316
8.3.27 Backup result state (configuration) ... 318
8.3.28 Backup result state (package execution) ... 321
8.3.29 Finish (configuration) ... 322
8.3.30 Finish (package execution) ... 324

8.4 Steps ... 326
8.4.1 Overview ... 326
8.4.2 Step tree structure ... 326
8.4.3 Edit step tree ... 328
8.4.4 Step properties ... 330
8.4.5 Script properties ... 335

8.4.6 Scripts ... 336
8.4.7 Step project ... 338
8.4.7.1 Overview ... 338
8.4.7.2 Exporting a step project from a selected step ... 339
8.4.7.3 Linking a step project ... 340
8.4.7.4 Removing a link to the step project ... 343
8.4.7.5 Refreshing the step project ... 344
8.4.7.6 Importing step attributes from the step project ... 345
8.4.7.7 Command line call ... 347

8.5 Managing and using configurations ... 350
8.5.1 Overview ... 350

8.6 Actions ... 351
8.6.1 Action properties ... 355
8.6.2 Action scripts ... 356
8.6.3 Copy ... 361
8.6.3.1 Description of the "Copy" action ... 361
8.6.3.2 Linking the directory structure ... 362
8.6.3.3 Notes on the "Copy" action ... 364
8.6.4 Change INI ... 364
8.6.4.1 Description of the "Change INI" action ... 364
8.6.4.2 Example action "Change INI" ... 367
8.6.5 Create with preprocessor ... 368
8.6.5.1 Description of the "Create with preprocessor" action ... 368
8.6.5.2 Examples of "Create with preprocessor" action ... 370
8.6.6 Delete ... 373
8.6.6.1 Description of the "Delete" action ... 373
8.6.6.2 Notes regarding the delete action ... 375
8.6.7 Execute ... 375
8.6.7.1 Description of "Execute" action ... 375
8.6.7.2 "Notes relating to the Execute action" ... 379

8.7 Script language ... 380

8.8 Reference list, system variables ... 380
8.8.1 General information about system variables ... 380
8.8.2 System variables for configuration ... 381
8.8.3 Enumerations for configuration ... 383
8.8.4 System variables for the dialogs ... 383
8.8.4.1 System variables for the dialogs ... 383
8.8.4.2 System variables for the Package configuration dialog ... 383
8.8.4.3 System variables for the Notes on the package dialog ... 384
8.8.4.4 System variables for Access to target system dialog ... 384
8.8.4.5 System variables for the select offline archive dialog ... 385
8.8.4.6 System variables for Backup initial state dialog ... 385
8.8.4.7 System variables for the PCU setup dialog ... 386
8.8.4.8 System variables for the CNC software dialog ... 386
8.8.4.9 System variables for the System configuration dialog ... 387
8.8.4.10 System variables for the SINAMICS topology dialog ... 387
8.8.4.11 System variables for the Version Display dialog ... 388
8.8.4.12 System variables for the Step tree dialog ... 388
8.8.4.13 System variables for Backup result state dialog ... 389
8.8.4.14 System variables for the Finish dialog ... 389

8.8.5 Enumerations for dialogs ... 389
8.8.5.1 Enumerations ... 389
8.8.5.2 Enumeration processing modes - ProcessMode ... 390
8.8.5.3 Enumeration selection of target system - AccessData ... 390
8.8.5.4 Enumeration installation modes - Mode ... 390
8.8.5.5 Enumeration, origin of NC data - NcSource ... 391
8.8.5.6 Enumeration source of the PLC data - PlcSource ... 391
8.8.5.7 Enumeration source of the DRV data - DrvSource ... 392
8.8.5.8 Enumeration, origin of HMI data - HmiSource ... 392
8.8.5.9 Enumeration, origin of system settings - SysSource ... 393
8.8.6 System variables for the steps ... 393
8.8.6.1 Overview ... 393
8.8.6.2 Query step ... 394
8.8.6.3 Activate step ... 394
8.8.6.4 Locking step activation ... 396
8.8.6.5 Collapse step ... 397
8.8.7 System variables for the environment ... 398
8.8.7.1 System variables for environment $Env ... 398
8.8.7.2 Enumeration, runtime environments - Runtimes ... 398

8.9 Forwarding a project/creating a package ... 399

8.10 Package execution ... 401
8.10.1 Requirements ... 401
8.10.2 Executing the package under Windows ... 403
8.10.2.1 Executing a standard package ... 403
8.10.2.2 Package execution dialogs ... 403

8.11 Command line call ... 404
8.11.1 Package deployment by Expert ... 404
8.11.2 Check run ... 407
8.11.3 Package execution ... 411

9 Create MyConfig - packages ... 415

9.1 Starting the package execution ... 415

9.2 Package execution dialog pages ... 415

9.3 Command line call ... 415

10 Create MyConfig - Topo ... 417

10.1 Introduction ... 417

10.2 Overview of user interface ... 419
10.2.1 Topo - Overview window ... 419
10.2.2 Topo - menu functions ... 420

10.3 Topo - operation ... 425
10.3.1 Creating a topology ... 425
10.3.2 Module catalog ... 433
10.3.3 DOList Editor ... 436
10.3.4 Manage DO list ... 439
10.3.4.1 Local DO list ... 442
10.3.4.2 Generating DO variables automatically ... 444
10.3.4.3 DO list ... 445
10.3.4.4 Predefinition of axis assignment for the EPLAN add-in ... 451

10.3.5 Topology comparison ... 451
10.3.6 Forwarding a topology ... 454
10.4 EPLAN Add-In ... 454
10.4.1 Inclusion of DRIVE-CLiQ topologies from EPLAN electric P8 ... 454
10.5 Command line call ... 455
10.5.1 Topo ... 455
10.6 SINAMICS device configuration ... 461
10.6.1 Production of modular machines ... 461
10.6.2 SINAMICS device configuration with Automatic configuration or user-specified topology ... 461
10.6.3 Drive data generation via "automatic configuration" ... 464
10.6.4 Drive data generation via "user-specified topology" ... 465
10.6.5 Assignment of the drive data using DO variables ... 465
10.6.6 The advantages of using comparison or user-specified topology ... 468
10.6.7 Configuration of the topology with Topo ... 470
10.6.8 Generation of the comparison topology using external software ... 471
11 FAQs ... 473
11.1 Introduction ... 473
11.2 Password handling for CMC V6.3 not supported ... 473
11.3 Restoring the original status of a USB memory stick ... 474
11.4 Important folders and files ... 475
11.5 Simulating a project based on offline archives ... 475
11.6 First encoder of a drive on the CU or downstream of the HUB ... 477
11.7 Parallel connection of Active Line Modules (ALM) ... 477
11.8 SINAMICS Starter: No access to drive after CMC package ... 478
11.9 Certain package names result in a Windows error message ... 478
A List of abbreviations ... 479
A.1 Abbreviations ... 479
Index ... 483

Introduction 1

1.1 About SINUMERIK

From simple, standardized CNC machines to premium modular machine designs – the SINUMERIK CNCs offer the right solution for all machine concepts. Whether for individual parts or mass production, simple or complex workpieces – SINUMERIK is the highly dynamic automation solution, integrated for all areas of production. From prototype construction and tool design to mold making, all the way to large-scale series production.

Visit our website for more information SINUMERIK (https://www.siemens.com/sinumerik).

1.2 About this documentation

Create MyConfig engineering software supports the machine manufacturer to create and run projects for automated production and commissioning of machines with SINUMERIK ONE controls. Upgrades of these controls are also configured by the OEM and then performed automatically at the end user.

The engineering software Create MyConfig comprises the 4 components Diff, Expert, Topo and Shield:

- **Diff** is a comparison software that supports you in the provision of modular data.
- **Expert** is a configuration software that supports you when creating a project. The project is run on the control in the form of an executable package.
- **Topo** is a software for configuring the topology and supports you in displaying and/or generating a comparison and/or target topology for the automated commissioning of the SINAMICS drive system.
  Topo is also used to support and preconfigure the axis-drive assignment.
- **Shield** is the execution software, which does not need to be individually installed. When deploying the project via Expert, an executable package containing the software component Shield, scripts, instructions and data is created. After starting the package, it is processed by Shield according to the configuration.

Independent tools

Create MyConfig also provides an autonomous tool that you can use without the engineering software:

- DOList Editor (Page 436)
  For editing reusable and cross-topology DO lists independently of Topo

Target group

This manual is intended for machine manufacturers and service technicians.

Benefits

As a result of the modular concept, Create MyConfig enables automatic series production and the upgrade of different machines of a series with one package.

In contrast to conventional commissioning, individual operations on the machine are safer, easier and faster to perform:

- The production time for machines with SINUMERIK ONE is significantly reduced.
- The upgrade time for end users is also reduced.
- A high level of reproducibility is achieved through automatic data loading.
- The quality of production and service is improved significantly.
- Less expert knowledge is required by the commissioning engineers.
- Modular data management for modular machines is supported at the machine manufacturer's.
- Modular machines can adapt to flexible SINAMICS topologies.
- The commissioning process is documented in a log file.

Standard scope

This documentation only describes the functionality of the standard version. This may differ from the scope of the functionality of the system that is actually supplied. Please refer to the ordering documentation only for the functionality of the supplied drive system.

It may be possible to execute other functions in the system which are not described in this documentation. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

For reasons of clarity, this documentation cannot include all of the detailed information on all product types. Further, this documentation cannot take into consideration every conceivable type of installation, operation and service/maintenance.

The machine manufacturer must document any additions or modifications they make to the product themselves.

Websites of third-party companies

This document may contain hyperlinks to third-party websites. Siemens is not responsible for and shall not be liable for these websites and their content. Siemens has no control over the information which appears on these websites and is not responsible for the content and information provided there. The user bears the risk for their use.

1.3 Documentation on the internet

Comprehensive documentation about the functions provided in SINUMERIK ONE Version 6.13 and higher is provided in the Documentation overview SINUMERIK ONE (https://support.industry.siemens.com/cs/ww/en/view/109768483).



You can display documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- User: Operating
- User: Programming
- Manufacturer/Service: Functions
- Manufacturer/Service: Hardware
- Manufacturer/Service: Configuration/Setup
- Manufacturer/Service: Safety Integrated
- Information and training
- Manufacturer/Service: SINAMICS

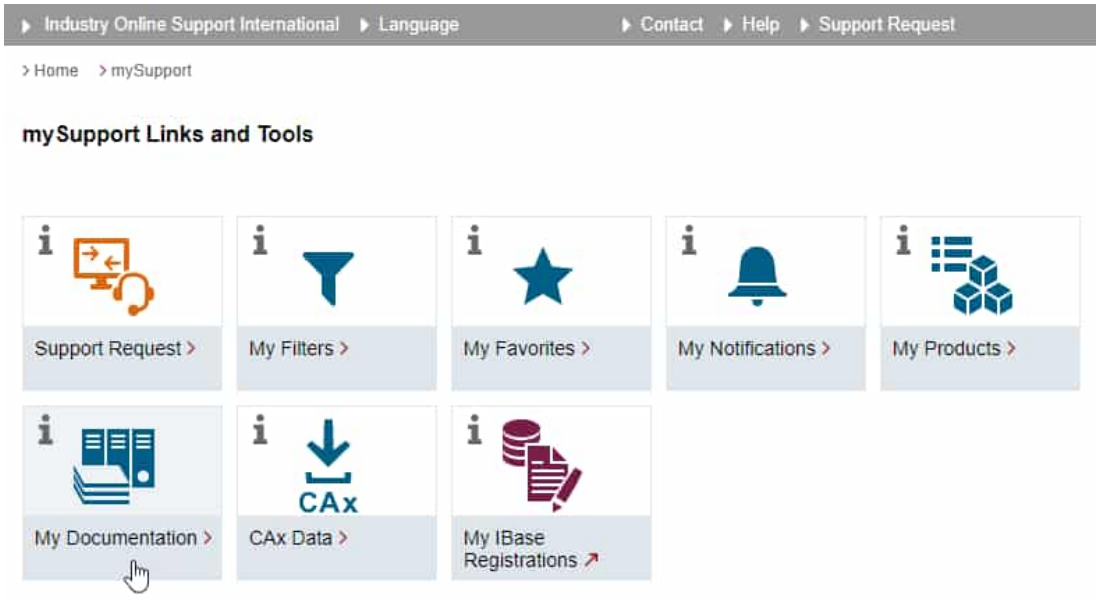
1.4 Feedback on the technical documentation

If you have any questions, suggestions or corrections regarding the technical documentation which is published in the Siemens Industry Online Support, use the link "Send feedback" link which appears at the end of the entry.

1.5 mySupport documentation

With the "mySupport documentation" web-based system you can compile your own individual documentation based on Siemens content, and adapt it for your own machine documentation.

To start the application, click on the "My Documentation" tile on the mySupport homepage (https://support.industry.siemens.com/cs/ww/en/my):



The configured manual can be exported in RTF, PDF or XML format.

Note

Siemens content that supports the mySupport documentation application can be identified by the presence of the "Configure" link.

1.6 Service and Support

Product support

You can find more information about products on the internet:

Product support (https://support.industry.siemens.com/cs/ww/en/)

The following is provided at this address:

- Up-to-date product information (product announcements)
- FAQs (frequently asked questions)
- Manuals
- Downloads
- Newsletters with the latest information about your products
- Global forum for information and best practice sharing between users and specialists
- Local contact persons via our Contacts at Siemens database (→ "Contact")
- Information about field services, repairs, spare parts, and much more (→ "Field Service")

Technical support

Country-specific telephone numbers for technical support are provided on the internet at address (https://support.industry.siemens.com/cs/ww/en/sc/4868) in the "Contact" area.

If you have any technical questions, please use the online form in the "Support Request" area.

Training

You can find information on SITRAIN at the following address (https://www.siemens.com/sitrain).
SITRAIN offers training courses for automation and drives products, systems and solutions from Siemens.

Siemens support on the go



With the award-winning "Siemens Industry Online Support" app, you can access more than 300,000 documents for Siemens Industry products – any time and from anywhere. The app can support you in areas including:

- Resolving problems when implementing a project
- Troubleshooting when faults develop
- Expanding a system or planning a new system

Furthermore, you have access to the Technical Forum and other articles from our experts:

- FAQs
- Application examples
- Manuals
- Certificates
- Product announcements and much more

The "Siemens Industry Online Support" app is available for Apple iOS and Android.

Data matrix code on the nameplate

The data matrix code on the nameplate contains the specific device data. This code can be read with a smartphone and technical information about the device displayed via the "Industry Online Support" mobile app.

1.7 Important product information

Using OpenSSL

This product can contain the following software:

- Software developed by the OpenSSL project for use in the OpenSSL toolkit
- Cryptographic software created by Eric Young.
- Software developed by Eric Young

You can find more information on the internet:

- OpenSSL (https://www.openssl.org)
- Cryptsoft (https://www.cryptsoft.com)

Compliance with the General Data Protection Regulation

Siemens observes standard data protection principles, in particular the data minimization rules (privacy by design).

For this product, this means:

The product does not process or store any personal data, only technical function data (e.g. time stamps). If the user links this data with other data (e.g. shift plans) or if he/she stores person-related data on the same data medium (e.g. hard disk), thus personalizing this data, he/she must ensure compliance with the applicable data protection stipulations.

Security instructions 2

2.1 Fundamental security instructions

2.1.1 General security instructions

⚠ WARNING

Danger to life if the safety instructions and residual risks are not observed

If the safety instructions and residual risks in the associated hardware documentation are not observed, 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

Malfunctions of the machine as a result of incorrect or changed parameter settings

As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death.

- Protect the parameterization against unauthorized access.
- Handle possible malfunctions by taking suitable measures, e.g. emergency stop or emergency off.

2.1.2 Warranty and liability for application examples

Application examples are not binding and do not claim to be complete regarding configuration, equipment or any eventuality which may arise. Application examples do not represent specific customer solutions, but are only intended to provide support for typical tasks.

As the user you yourself are responsible for ensuring that the products described are operated correctly. Application examples do not relieve you of your responsibility for safe handling when using, installing, operating and maintaining the equipment.

2.1.3 Security information

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

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

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

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

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

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under
https://www.siemens.com/industrialsecurity (https://new.siemens.com/global/en/products/services/cert.html#Subscriptions).

Further information is provided on the Internet:

Industrial Security Configuration Manual (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ WARNING

Unsafe operating states resulting from software manipulation

Software manipulations, e.g. viruses, Trojans, or worms, can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

- Keep the software up to date.
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
- Make sure that you include all installed products into the holistic industrial security concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.
- On completion of commissioning, check all security-related settings.

2.2 Product-specific security instructions

2.2.1 Password handling for CMC V6.3 not supported

Take the following note for CMC V6.3 into account:

Note

PLC projects with activated PLC protection levels or F-passwords are not supported

PLC protection levels or F-passwords (TIA Portal: "password for modifiying safety program") are not supported for CMC 6.3 and CNC-SW6.15 and lower.

For filling using CMC V6.3 and CNC-SW6.15 and lower, do not use PLC projects with active PLC protection levels and F-passwords.

Instead, use the password-protected HMI Operate access levels "Manufacturer" or "Customer" to protect the safety configuration in the control.

If the control currently has a loaded PLC project which is protected by the passwords as described, CMC V6.3 can neither create a backup of the PLC portion nor update the current PLC configuration.

In the documentation SIMATIC Programming and Operating Manual "SIMATIC Safety - Configuring and Programming" (https://support.industry.siemens.com/cs/de/en/view/54110126) for TIA Portal V17, the section "Access protection" defines administering a PLC access level and the definition of an F-password in the "Safety Administration Editor" in order to protect a PLC safety configuration from unauthorized modification.

Within the framework of the SINUMERIK ONE STEP 7 Toolbox V17, you can also apply these measures to the integrated PLCs of the SINUMERIK ONE. Exporting a respective DSF-PLC archive from the TIA Portal is possible.

In SINUMERIK V6.15, you can also read in an access-protected DSF-PLC archive directly at the control via HMI Operate while taking the defined passwords for PLC protection levels and F-password into account.

2.2.2 Administer the computer according to general security guidelines

Note

Carefully observe the appropriate security guidelines for computers

For the computers used in the workflow within the context of Create MyConfig, the network environment used as well as operating system, carefully ensure that the general security guidelines are applied when it comes to administering the system.

These include, for example:

- Activated user logon
- Patched operating system
- Firewall
- Virus scanner

Additional information is provided in the following documentation:

SINUMERIK / SIMOTION / SINAMICS Motion Control Industrial Security (https://support.industry.siemens.com/cs/ww/en/view/108862708)

2.2.3 CMC Shield

Note

Network protocol SSH / archive exchange format DSF

The CMC Shield program uses the "SSH" network protocol to access the NCU or IPC.

The SSH protocol is only installed as standard on the NCU and on a computer with the Windows 10 operating system.

Use the archive exchange format "DSF" for exchanging data between other systems/participants.

Create MyESW

3

The automated engineering and service workflow "Create MyESW" meets the machine tool manufacturer's requirement during automation for commissioning and the upstream engineering workflow.

A completely automated workflow is thus achieved for series machines, modular machines and series machines equipped with options. At the press of a button, essentially free of manual entries - from customer order through to commissioning.

The engineering workflow, as well as commissioning, can be tested in the office on a PC via "Create MyVirtual Machine" (CMVM).

The engineering workflow only functions for SINUMERIK ONE.

Create MyConfig (CMC)

CMC is integrated in the engineering workflow:

- OEM project
  Data, for example from EPLAN or the TIA Portal, can be imported into the CMC project
- Machine installation package
  The CMC package for installation is ready for commissioning and operators

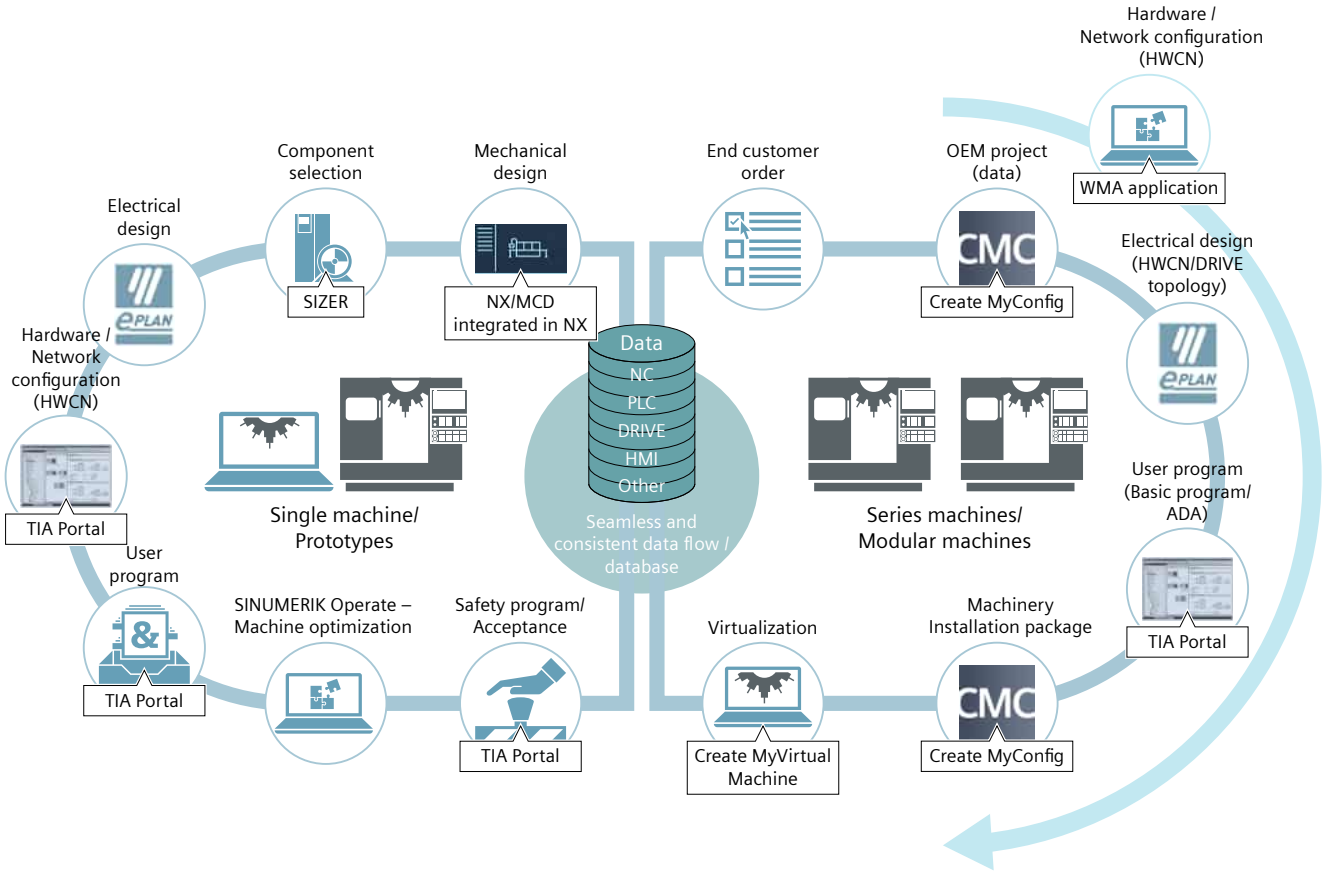


Figure 3-1 Overview - Seamless and consistent data flow

Note

Create MyESW - Engineering & Service Workflow

A detailed description of Create MyESW is provided with the "Create MyESW - Engineering & Service Workflow" System Manual.

You can find the documentation in SIOS under:

https://support.industry.siemens.com/cs/de/en/view/109801013 (https://support.industry.siemens.com/cs/de/en/view/109801013)

Installation

4

4.1 Overview

You have 2 options for installing engineering software "Create MyConfig" on your PC:

- Installation using the installation wizard
  The CMC package is installed using what is called the General setup.
  The general setup involves the installation of Create MyConfig - and the installation of Application License Managers (ALM).
  Additional information can be found in the following sections:
  - User-guided setup (Page 24)
  - Performing the automated installation (Page 24)
- Installation via command line
  It can also be automatically installed without any user interaction.
  An "OPFILE.txt" control file can be transferred as argument when starting the setup.
  In this control file, you can preconfigure the setup workflow. This means that it is no longer necessary to manually work through the setup dialogs. The following sections describe installation via the command line.

4.2 Hardware requirements

A PC/PG must meet the following requirements for the installation and operation of Create MyConfig (CMC).

- The installation drive should have approx. 900 MB of free memory.
- Additional memory is required to create your own projects.
- All graphics-capable printers supported by the operating system can be used.
- A USB data medium with or without Linux boot system (USB stick, USB hard disk, etc.) or a network connection is required to perform a new installation of the NCU.
- For a new installation, take into account the preconditions described in the release notes for the particular SINUMERIK version.

4.3 Software requirements

Create MyConfig (CMC) is a 32/64-bit application and requires one of the following software platforms:

PG/PC	MS Windows 10 (64-bit) Professional, Enterprise
IPC 427E	Windows 10 IoT Enterprise 2016 LTSB (64-bit) with installed PCU-Basesoftware
IPC 477E	Windows 10 IoT Enterprise 2016 LTSB (64-bit) with installed PCU-Basesoftware

Furthermore, CMC requires the following installed products:

- Web browser
- Acrobat Reader or a comparable tool for displaying PDF documents

4.4 User-guided setup

Create MyConfig is installed via a user-guided setup (general setup).

The Automation License Manager (ALM) which is required for the script execution is also installed via the user-guided setup.

To do this, start "Setup.exe" in the root directory of the DVD:

`LW:\[DVD setup path]\.`

The CMC setup files can now be found in the following directories:

For the 64-bit version: `LW:\[DVD setup path]\setup\CMC\setup\x64`

For the 32-bit version: `LW:\[DVD setup path]\setup\CMC\setup\x86`

Note

SINUMERIK ONE - CMC 6.3

- You can install and use CMC 6.3 in parallel with the previous versions CMC 4.5, 4.6 HF1, 4.7 SP2 HF1, CMC 4.8 SP1, V6.0, V6.1 and V6.2.
- You can also install individual components of CMC 6.3.

4.5 Performing the automated installation

Information about automatically installing Create MyConfig is provided in the PDF supplied:

- PEAutoInstalldeDE.pdf
  Programming and Operating Manual "Performing an automated installation"

4.6 Extensions to CMC

The following add-ons are available for CMC as of Version 6.0:

- Expansion for CMC/Topo:
  - Create MyConfig - ECAD IMPORT product id MCS10201
- Expansion for CMC/Diff:
  - Create My Config - SCRIPTING product id MCS10202

4.7 Starting Create MyConfig

You can start the Diff, Expert and Topo components via the Windows Start menu or the links on the desktop.

You can open project files or components directly in the assigned component by double-clicking.

Uninstallation

5

Create MyConfig can be uninstalled via the standard Windows mechanisms or by restarting the setup program.

Note

User-specific data is retained

After the uninstallation, user-specific data of CMC may be retained under the following paths, and may be reused after a new installation.

```
%ProgramData%\Siemens\Automation\SINUMERIK\Create MyConfig 6.3
```

```
%LocalAppData%\Siemens
```

Create MyConfig - CMC script

6

6.1 Introduction

Note

Documentation structure: Operating Manual / Online Help

The contents of Section **CMC-Skript** are identical in the Operating Manual and the online help.

Under CMC script, you can find all the information on:

- Integrated Script Editor (Expert)
- CMC script language
- Script libraries

The information can involve general Create MyConfig or specifically the **Diff** or **Expert** software.

Navigate to the relevant information via notes or links.

6.2 Script Editor

The "Script Editor" is integrated in Expert and is automatically opened in the home area when you double-click a script for editing.

Representation of the integrated Script Editor in Expert

For each script a separate tab is opened in Expert in the home area.

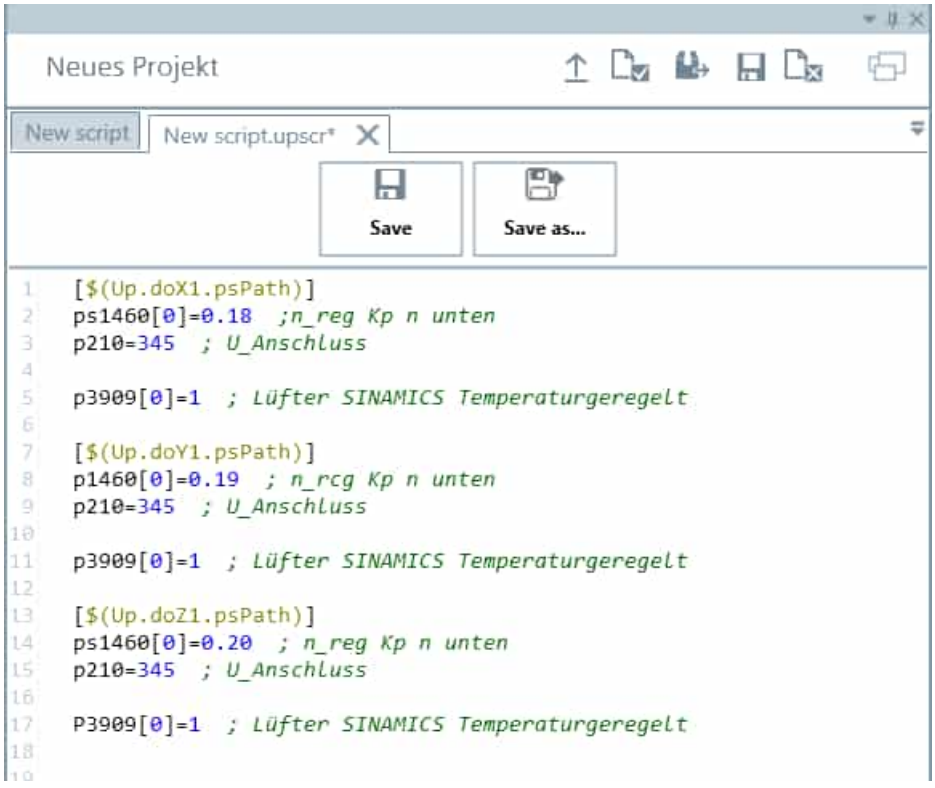


Figure 6-1 Example illustration: integrated Script Editor

6.3 #include - script

#include "<path>"

The `#include` instruction is a deployment directive.

These directives can be called in all scripts, dialog scripts and action scripts; only "CMC scripts in Diff" are excluded.

You can use relative paths in include statements in scripts.

You specify the file path of the external file `<path>` to be included in relative or absolute terms.

Note that the paths refer to the following folders depending on the script type:

- For dialog scripts and scripts in the step tree, the relative path refers to the project folder.
- For linked dialog scripts/scripts in the step tree, the relative path refers to the folder where the script is saved.
- For action scripts, the relative path refers to the folder where the action is saved.
  - This also applies to linked actions
  - This also applies to the order of an action of the type "Create with preprocessor"
  - If the order in an action of the type "Create with preprocessor" is linked, the relative path refers to the order folder.

The path can contain Windows environment variables, e.g. `%SUB_PATH%`.

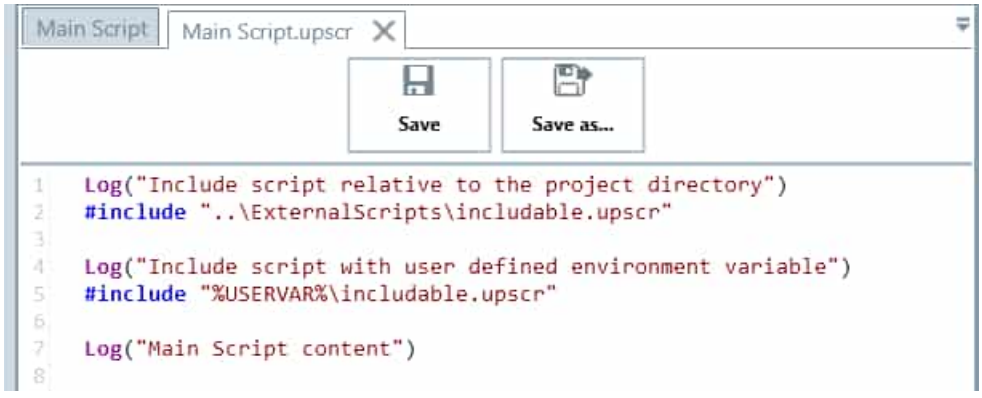


Figure 6-2 Example of #include script

The validity of the include statement is checked:

- The files to be included are also checked by the check run ("Review project" button).
- When a package is deployed, the contents of the external file are taken over in the script instead of the include statement.

6.4 Locations where the script languages is used

Locations where the script languages is used

The section "Create MyConfig - CMC script" describes a simple programming language that is based on known elements of other programming languages.

You can create scripts for dialogs and steps, action scripts for actions of the type "Copy", "Change INI", "Create with preprocessor", "Delete", "Execute", and for Diff.

The "Script Editor" integrated in Expert supports you in creating and editing scripts by:

- Syntax highlighting
- Extensive shortcut menu

You use scripts in:

- Dialogs ("dialog scripts (Page 242)")
- Steps (Expert > Steps (Page 326))
- Actions (Expert > Actions > "Action scripts (Page 356)")
- Change orders: Examples of an action script (Expert > Actions > "Change INI (Page 367)" / "Create with preprocessor (Page 370)")
- Diff ("CMC scripts in Diff (Page 151)")

Dialogs

You can have scripts executed at fixed times on almost all dialog pages.

You can find more information under "Dialog scripts (Page 242)".

Steps

You can save scripts for each step in the step tree, which are executed when executing the step.

Using scripts, you can control the current step and all subsequent steps.

More information is provided under Expert in Section "Steps" > "Scripts" (Page 336)

Actions

For each action, you can save an action script; this is first executed before the associated action is executed. This allows you to control the execution of the Action.

More information is provided in Expert in the section under "Actions (Page 351)".

Diff

You can use scripts in Diff.

More information can be found at:

- "CMC scripts in Diff (Page 151)" in the Diff online help or in the section of the same name in the Operating Manual
- "System variables - CMC scripts in Diff (Page 34)" in the CMC script online help or in the section of the same name in the Operating Manual

Note that you require a license for "CMC scripts in Diff":

- Create MyConfig - SCRIPTING MCS10202

6.5 Color coding of language elements

For better orientation, important language elements are color-coded in the "Script Editor". The remaining program elements are displayed black and unformatted.

In the following you will find the meaning of the color markings with examples.

Table 6-1 Meaning of the color markings

Program element	Color	Formatting	Example
Comment	Dark green	Italic	`; Comment OK`

Keyword	Blue	Bold	`if Up.doX != Null`
			`  Msg("doX set")`
			`  else`
			`  Msg("doX not set")`
			`  endif`
			
			`while Up.bit_idx < 5`
			`  Up.bit_value *= 2`
			`  Up.bit_idx += 1`
			`  endwhile`

Sections	Black	Bold	`Chandata(1)`
			`[C1]`
			`[B3_S3_DO1]`

Decimal numbers	Blue	-	`DEZ = 123 ;`
			`DEZ_GP = 123.45`
			`DEZ_EX = 123ex-1`
			`VER = 2.6.1`

Binary numbers
Hexadecimal numbers
Binector/Connector (BICO technology)
	Violet	-	`HEX_NC = 'H123'`
			`HEX_0x = 0x123`
			`BIN_NC = 'B0101'`
			`BICO = 3:1.23`

STRING	Chestnut brown	-	`STR = "text"`

Functions	Violet
	Chestnut brown
		Bold	`BOOL = Exists(NCU, "test")`
			`VER = Version(NCU, "hjiji9")`
			`VAR = Input("trala")`

Complex variables whose value is **not** interpreted byCMC	Yellow/orange	-	`$P_UIFR[1] = CTRANS(x, 0.5, y 0.2)`
Replacement operator	Kaki	-	`Up.lin_3_$(Up.dotmp)=1`

6.6 System variables - CMC scripts in Diff

6.6.1 Overview of system variables

System variables for CMC scripts in Diff are described in the following table, and examples of their use given.

Table 6-2 System variables in CMC scripts in Diff

System variables	Type	Change	Description	Example
Up.$Env.Batch-Mode (Page 35)	BOOL	readOnly	This system variable indicates as to whether the script runs in the "BatchMode", i.e. was called from a command line.	`If Up.$Env.batchMode == false`
Up.$Script.Script-File (Page 35)	STRING		This system variable contains the actual path of the CMC script file that is being run.	`If Up.$Script.ScriptFile != "P:\Manufact\Diff-Script \myscript.upscr"`
Up.$Script.ArcFile (Page 35)	STRING		**Start via command line**: This variable contains the path of the archive that was assigned using switch `"/outarc"`. **Start via user interface**: This variable contains the path of the archive to which the script is applied.	`If Up.$Script.ArcFile != "C:\Temp\mync.arc"`
Up.$Script.Result (Page 36)	STRING	Can be read and written to.	All files that you assign to these system variables in the CMC script are transferred from Diff to the instance that calls the CMC script. **Start via user interface**: The files are opened after execution. **Start via command line**: After execution, the files are transferred corresponding to the arguments at switch `/result`.	`Up.$Script.Result = "./myResult.txt"`
Up.$Script.Edit-Mode (Page 38)	BOOL	readOnly	This system variable signals as to whether Diff is in the Edit mode.	`If UP.$Script.EditMode != true`

A more detailed description and examples for using the system variables in CMC scripts is provided in the section under "Using system variables (Page 34)".

6.6.2 Using system variables

System variables for CMC scripts in Diff are described in this Section, and examples of their use given.

6.6.2.1 Up.$Env.BatchMode

Command line execution of CMC scripts in Diff is always realized in the "Batch mode".

Using system variable `Up.$Env.BatchMode`, you can interrogate as to whether the CMC script is in the "Batch mode", i.e. the CMC script was started via the command line.

Example

If you do not wish to call any dialogs while running the script in the "Batch mode", then make the call dependent on the dialog of variables `Up.$Env.BatchMode`.

```
If Up.$Env.BatchMode == false
 Msg("The CMC script was started from the user interface of Diff.")
EndIf
```

6.6.2.2 Up.$Script.ScriptFile

Variable `Up.$Script.ScriptFile` contains the absolute path of the script file that is executed.

Using this variable, you check whether the correct CMC script was started.

Example

```
If Up.$Script.ScriptFile != "P:\Manufact\Diff_Script\myscript.upscr"
 Error("The incorrect CMC script was started! It is possible that
 this CMC script is not current!")
EndIf
```

Note

Case sensitivity of paths

It is important that you take into account the case sensitivity if you perform a string comparison of paths. Under Windows, this is not relevant in the file system.

When comparing string values in the CMC script you must take into account the case sensitivity.

6.6.2.3 Up.$Script.ArcFile

For command line execution, the system variable contains the absolute path of the result archive, which was specified using the command line switch `/outarc`.

When starting the CMC script via the user interface, the variable contains the absolute path of the (result) archive on which the CMC script is applied.

Example

```
MSG("The archive '$(Up.$Script.ArcFile)' is manipulated by the CMC
script.")
```

6.6.2.4 Up.$Script.Result

You require this system variable for data export of your CMC script in Diff. All files that you assign to these system variables in the CMC script are transferred from Diff to the instance that calls the script.

When starting the CMC script from the user interface of the Diff, the files are opened after the script has been run and transferred to the user. The user saves the files with "Save as" in the linked standard application at the required location.

When starting the CMC script from the command line, after the script has been executed, the files are saved or transferred corresponding to the argument at switch `/result`.

You only require system variable `Up.$Script.Result` if these two conditions are satisfied:

- When using procedures that generate files,
  File operations, e.g. TraceToFile()
- When using relative paths in these procedures,
  e.g. because the CMC script is to run on various configured computers

Example 1 - exporting a file

Content CMC script `"myScript.upscr"`:

```
TraceToFile(".\myTrace.txt", "my trace output")
Up.$Script.Result = ".\myTrace.txt"
```

Content batch script (CMD) for command line:

```
@echo off
echo script is running ...
set cmc_diff="C:\Program Files\Siemens\Automation\SINUMERIK\Create
MyConfig 6.3\Diff.exe"
set script="myScript.upscr"
set arc_file="mync.arc"
set out_arc="mync_new.arc"
set script_log="myScript.html"
set result="C:\Temp\myResult.txt"

%cmc_diff% /script %skript% /inarc %arc_file% /outarc %out_arc% /
result %result% /scriptlog %script_log% /log cmdLog.xml
echo Errorlevel is: %Errorlevel%
Pause
```

Result after executing the batch and CMC script:

After executing the CMC script using the command line call, the content of file "myTrace.txt" is located in file "C:\Temp\myResult.txt".

After executing the CMC script when started at the Diff user interface, the content of file "myTrace.txt" is located in the editor. You save the file using "Save as" at a location that you can arbitrarily select.

Example 2 - exporting several files

When exporting several files, use a semicolon to separate the files. Assign the files as string variable to system variable `Up.$Script.Result`.

The path of a folder is saved in the batch script for the argument of switch `/result`. The exported files are saved in this folder.

Content CMC script `"myScript.upscr"`:

```
TraceToFile(".\myTrace.txt", "my trace output")
FileCopy(ARC, "NC/initial.ini", RTS, ".\initial.ini")
Up.$Script.Result = ".\TraceToFile.txt;.\initial.ini"
```

Alternatively, you can also assign the files using package variables and concatenate operator or using package variables and replacement operator.

Using package variables and concatenate operator:

```
Up.File1 = ".\myTrace.txt"
Up.File2 = ".\initial.ini"
Up.$Script.Result = Up.File1 << ";" << Up.File2
```

Using package variables and replacement operator:

```
Up.$Script.Result = "$(Up.File1) ; $(Up.File2)"
```

Content batch script (CMD) for command line:

```
@echo off
echo script is running ...
set cmc_diff="C:\Program Files\Siemens\Automation\SINUMERIK\Create
MyConfig 6.3\Diff.exe"
set script="myScript.upscr"
set arc_file="mync.arc"
set out_arc="mync_new.arc"
set script_log="myScript.html"
set result="C:\Temp\myResult"

%cmc_diff% /script %skript% /inarc %arc_file% /outarc %out_arc% /
result %result% /scriptlog %script_log% /log cmdLog.xml
```

```
echo Errorlevel ist: %Errorlevel%
Pause
```

Result after executing the batch and CMC script:

After executing the CMC script using a command line call, files "myTrace.txt" and "nitial.ini" are located in folder "C:\Temp\myResult". If the folder does not exist, then Diff automatically creates the folder.

After executing the CMC script when started at the Diff user interface, files "myTrace.txt" and "nitial.ini" are opened, each in one editor. You save the files using "Save as" at a location that you can arbitrarily select.

6.6.2.5 Up.$Script.EditMode

By interrogating these system variables, you can identify as to whether Diff is in the Edit mode.

Example

```
If UP.$Script.EditMode != true
 Error("You have forgotten to activate the Edit mode!")
Endif
```

Note

Starting the CMC script via the command line

When starting a CMC script via the command line, Diff is always automatically in the Edit mode. For command line execution, system variable UP.$Script.EditMode always has the value true.

6.7 Script libraries

Frequently used script blocks can be stored in libraries as procedures or functions so that these script blocks are available for several projects or CMC scripts in Diff.

Notes in CMC scripts in Diff

Note

Using script libraries in CMC scripts in Diff

You have the following ways to publish your own libraries if you want to use your own functions and procedures from your script libraries in CMC:

- Store your libraries next to the CMC script for Diff. In this way, all scripts are in the same folder.
- Store the libraries in a path (folder) of the environment variable `%UP_LIB_PATH%`.
- Include your archive path in the variable.

Environment variable `%UP_LIB_PATH%` is only relevant when executing CMC scripts in Diff.

Note

Identifying script libraries

When starting a CMC script in Diff via the user interface or a command line call, the script libraries are read-in one after the other in the following sequence:

- Libraries next to the script
- 1st path in `%UP_LIB_PATH%`
- 2nd path in `%UP_LIB_PATH%`
- ...
- n. path in `%UP_LIB_PATH%`

If the definition of a procedure or function is available in more than one script library (multiple definition of the same function or procedure), a corresponding error message is generated when the CMC script is started using the Diff user interface. You then have the option of continuing or cancelling.

When the CMC script is started via the command line, the error message is written to the logbook. Shield uses the first definition found in accordance with the search sequence described above.

6.8 CMC standard script libraries

6.8.1 CMC standard script libraries

The CMC standard script libraries are provided by Create MyConfig:

- SINAMICS (Page 40)
- Utilities (Page 40)

You can find standard script libraries under the following path:

- C:\Program Files\Siemens\Automation\SINUMERIK\Create MyConfig 6.3\UpLib

Note

More information

Detailed information on the functions of the CMC standard script libraries is provided in Section "From library (Page 103)".

6.8.2 SINAMICS library

Function/procedure of the "SINAMICS" library

The `SinamicsTecActivate()` and the `SinamicsTecDeactivate()` procedures are available in the "SINAMICS" library. The procedures activate or deactivate technological functions in the drive.

6.8.3 Utilities library

Functions of the "Utilities" library

The `IsBit()`, `SetBit()` and `ClrBit()` functions are available in the "Utilities" library. You can manipulate or query the individual bits of any variables with these functions.

6.9 Example exporting archive data

A CMC script to export a selection of NC axis machine data could, in practice, look like the following.

Content, script "Read_Axis_Values.upscr"

```
Up.myWorkingDir = "."
Up.myResult = ""

; Writes the first line of the result
Up.myResult = "Example for exporting data using a CMC script:" << '\n'

Up.nAxis = 1
While(Up.nAxis < 35)
  Up.myAxisIdx = Up.nAxis - 1
  Up.myResult = Up.myResult << " Information about the axis $(Up.nAxis)" << '\n'
```

```
  Up.myValue = NC[C1].$MN_AXCONF_MACHAX_NAME_TAB[$(Up.myAxisIdx)]
  If (Up.myValue != null);Does the data have a value, i.e. is the data available?
    Up.myResult = Up.myResult << " $MN_AXCONF_MACHAX_NAME_TAB=$(Up.myValue)" <<
    '\n'
  EndIf
  Up.myValue = NC[C1].$MA_CTRLOUT_MODULE_NR[0,AX$(Up.nAxis)]
  If (Up.myValue != null)
    Up.myResult = Up.myResult << " $MA_CTRLOUT_MODULE_NR[0]=$(Up.myValue)" << '\n'
  EndIf

  Up.nAxis += 1
EndWhile

; Configure path to the result file and write content
Up.$Script.Result = Up.myWorkingDir << "/myResult.txt"
FileWrite(RTS, Up.$Script.Result, Up.myResult)
```

Content batch script (CMD) for command line

```
@echo off
echo script is running ...

set cmc_diff="C:\Program Files\Siemens\Automation\SINUMERIK\Create MyConfig
6.3\Diff.exe"
set skript=" Read_Axis_Values.upscr"
set arc_file="mync.arc"
set script_log=" Read_Axis_Values.html"
set result="C:\Temp\myResult.txt"

%cmc_diff% /script %skript% /inarc %arc_file% /result %result% /scriptlog
%script_log% /log cmdLog.xml
echo Errorlevel is: %Errorlevel%
Pause
```

Result and response after executing the batch and CMC script

After executing the CMC script using the command line call, the exported data is located in file "C:\Temp\myResult.txt".

After executing the CMC script when started at the Diff user interface, the content of file "myResult.txt" is located in the editor. You can save the file using "Save as" at a location that you can arbitrarily select.

Content of file "myResult.txt" abbreviated:

Example for exporting data using a CMC script:

```
  Information about axis 1
    $MN_AXCONF_MACHAX_NAME_TAB=X1
    $MA_CTRLOUT_MODULE_NR[0]=1
  Information about axis 2
    $MN_AXCONF_MACHAX_NAME_TAB=Y1
    $MA_CTRLOUT_MODULE_NR[0]=2
  Information about axis 3
    $MN_AXCONF_MACHAX_NAME_TAB=Z1
    $MA_CTRLOUT_MODULE_NR[0]=3
  Information about axis 4
    $MN_AXCONF_MACHAX_NAME_TAB=A1
  Information about axis 5
    $MN_AXCONF_MACHAX_NAME_TAB=B1
...
  Information about axis 9
    $MN_AXCONF_MACHAX_NAME_TAB=
  Information about axis 10
    $MN_AXCONF_MACHAX_NAME_TAB=
...
  Information about axis 32
  Information about axis 33
  Information about axis 34
```

6.10 Example, manipulating archive data

A CMC script to manipulate NC axis machine data could, in practice, look like the following.

The precondition is that no axis gaps are generated and the data sets actually exist.

You must perform an appropriate check and integrate the appropriate check if axis gaps exist, and/or data sets are missing.

Content, script "Manipu_NC_Axis_Values.upscr"

```
If UP.$Script.EditMode != true
  Error("You have forgotten to activate the Edit mode!")
EndIf
NC[C1] ;Fully qualified section
Up.nAxis = 1
While(Up.nAxis < $ON_NUM_AXES_IN_SYSTEM+1)
```

```
  Up.nDataSet = 0
  While(Up.nDataSet < 6);write value for all 5 data sets
    If $MA_IS_ROT_AX[AX$(Up.nAxis)] == 0 ;axis is a linear axis
      N36200 $MA_AX_VELO_LIMIT[$(Up.nDataSet),AX$(Up.nAxis)] = $MA_MAX_AX_VELO[AX$
      (Up.nAxis)] * 1.12
    Else ;axis is a spindle
      N36200 $MA_AX_VELO_LIMIT[$(Up.nDataSet),AX$(Up.nAxis)] =
      $MA_GEAR_STEP_MAX_VELO_LIMIT[$(Up.nDataSet),AX$(Up.nAxis)] * 1.13
    EndIf
    Up.nDataSet += 1
  EndWhile
  ;all data without data set
  If $MA_IS_ROT_AX[AX$(Up.nAxis)] == 0 ;axis is a linear axis
    N36060 $MA_STANDSTILL_VELO_TOL[AX$(Up.nAxis)]=4
  Else ;axis is a spindle
    N36060 $MA_STANDSTILL_VELO_TOL[AX$(Up.nAxis)]=0.012
  EndIf
  N36030 $MA_STANDSTILL_POS_TOL[AX$(Up.nAxis)]=3
  N36040 $MA_STANDSTILL_DELAY_TIME[AX$(Up.nAxis)]=0.5
  Up.nAxis += 1
EndWhile
```

Content batch script (CMD) for command line

```
@echo off
echo script is running ...

set cmc_diff="C:\Program Files\Siemens\Automation\SINUMERIK\Create MyConfig
6.3\Diff.exe"
set skript="Manipu_NC_Axis_Values.upscr"
set arc_file="mync.arc"
set out_arc="mync_new.arc"
set script_log="Manipu_NC_Axis_Values.html"

%cmc_diff% /script %skript% /inarc %arc_file% /outarc %out_arc% /scriptlog
%script_log% /log cmdLog.xml
echo Errorlevel is: %Errorlevel%
Pause
```

Result and response after executing the batch and CMC script

After executing the CMC script using a command line call, output archive "mync.arc" is saved with the manipulated data as "mync_new.arc".

After executing the CMC script when started at the Diff user interface, the data of output archive "mync.arc" is manipulated. Archive "mync.arc" is in an unsaved state, marked by a star symbol "*". You save the archive using "Save as".

6.11 Package variables

Package variables - Up.<identifier>

Package variables are variables that have significance only within a package. The control does not recognize package variables. Therefore, package variables are not transferred to the controller.

The package variables comprise an identifier, preceded by the "Up." prefix.

The characters that are also permissible in the NC for <identifier> are permissible for the identifier. In principle, the same rules therefore apply as with the NC, e.g. no umlauts, no spaces.

Depending on their definition, package variables are classified according to the following categories:

Category	Defined by	Value allocation by
Script	User texts in scripts	User
DO (drive object)	User in "SINAMICS topology" dialog page or via the DOVar function	System
System	The system using the package configuration	System

Table 6-3 Package variable types used in CMC

Variable	Syntax *	Description
Script variable	Up.<Name>	Value of an Up variable
DO variable	Up.<Name>.<Property>	Property value of a DO variable (drive variable)
System variable	Up.$<SysName>.<Property>	Property value of a system variable

* Package variables are identified by the "Up." prefix.

All package variables are defined in the same namespace. Consequently, unique names ("Name") must be used throughout all categories. Package variables are accessed using the "Up." prefix.

You can use package variables in:

- Dialog scripts
- Action scripts
- Scripts
- User interface texts
- Functions
- Notes
- Replacement operators
  You can find more information at "Special operators (Page 61)".
- In all script areas
  You can find more information at "Data accesses (Page 50)".

A variable name "Name" that can be entered by the user must start with a letter or an underscore and comprises the characters `A-Z_0-9`. Uppercase/lowercase is not relevant.

Package variables that receive their value from the system can only be accessed in read-only mode.

6.11.1 Script variables

Script variables - `Up.<Name>`

You create script variables using instructions in the form `Up.<Name> = <Value>`, using a unique name `<Name>` and immediate value initialization `<Value>`.

Script variables can be read and written in all scripts.

More information is provided in Section "Package variables (Page 44)".

In change orders in "Preprocessor script" mode, script variables can **only** be read in control structures, writing is not possible.

Examples

```
Up.M1=100.123
Up.xx_1= R[7]
Up.NX_BOOL=true
Up.Kanal="channel_1"
R[5] = Up.M1
If Up.Number == 123...
```

6.11.2 DO variables

DO variables - `Up.<doVar>`

The DO variables that can be used in the package (drive object variables) are defined in the topology on dialog page "SINAMICS topology (configuration) (Page 299)".

The variables are instantiated and the property values assigned when executing on this dialog page by allocating to the existing drive components.

In scripts, you read access properties of DO variables using "`Up.<Name>.<Property>`".

Table 6-4 The following properties are available for a DO variable with name "doVar".

Up.doVar != null	
	Supplies "true" if the DO variable was assigned to a drive component at runtime.
Up.doVar.doNr	
	Supplies the drive object number (Drive Object) **Note**: Configured renumbering of the drive objects is performed before the variable assignment
Up.doVar.slaveNr	
	Supplies the number of the higher-level slave object
Up.doVar.busNr	
	Supplies the number of the higher-level bus object
Up.doVar.psPath	
	Supplies the path to the PS file, e.g. `B3_S3_PS31`
Up.doVar.dpSlot	
	Supplies the PROFIBUS location number, starting with "1" (only for servo of the NCU and NX)
Up.doVar.doType	
	Supplies the drive object type. Possible values lie between "0" and "600". The valid values correspond to the values set in SINAMICS at the Control Unit in parameter "drive objects type / DO type" P0107. **Examples** Possible values for "Up.doVar.doType" are: • The value "3" corresponds to DO type "SINAMICS I". • The value "4" corresponds to DO type "SINAMICS NX/CX32". • The value "10" corresponds to DO type "ACTIVE INFEED CONTROL". • The value "11" corresponds to DO type "SERVO". • The value "254" corresponds to DO type "CU-LINK".

6.11.3 System variables

System variables `Up.$<Sysname>`

System variables are package variables that provide information on the package configuration and runtime environment. System variables are identified by a `'$'` prefixed to the name `<Sysname>`.

System variables are assigned as follows:

- Up.$Dialog - "Dialogs (Page 383)" (Expert > Reference list system variables > System variables for the dialogs)
- UP.$Env - "System variables for the environment (Page 398)" (Expert > Reference list system variables > System variables for the environment)

Note

A reference list for the system variables is provided in Expert in Section "Reference list system variables (Page 380)".

6.12 Functions and procedures

Comparison of functions and procedures

Functions and procedures are subprograms, which are called up together with parameters. The parameters are transferred in the form of arguments.

The difference between functions and procedures is that procedures do not supply a direct return value.

Functions supply one direct return value, e.g. a value that the operator entered.

You can only use functions at the following locations:

- Within comparisons
- As argument of other functions or procedures
- To the right of a "="

Special issue relating to procedures:

Procedures must be located alone in one line.

You can use the following as arguments for functions and procedures:

- Constant value
- Variable
- Up variables
- System variables

- Local or global area data (control data from the archive)
- Functions

Excluded are only those arguments that contain an ID parameter. An ID parameter is a string without exclamation marks, e.g. RTS, ARC, NCU, PCU for argument `<area>` of function `FileExist()`.

Call examples

Call examples for functions and procedures

Functions

```
$MC_CHAN_NAME=InputText(Up.Label,$MC_CHAN_NAME) ;function with variable arguments in the assignment
```

```
If Input("number of tools") < 600 ; function in condition
```

```
Up.small_magazine = Input("number of tools") < 600 ;function in condition
```

```
If FileEXIST(NCU, "/siemens/versions.xml") ;function in condition
```

Procedures

```
MSG("do not forget magazine selection!") ;procedure with constant argument
```

```
TraceToFile(Up.file,Up.String) ;procedure with Up variable as argument
```

```
FileWrite(RTS,"MyText.txt","line 1"<<'\r'<<'\n'<<"line 2"<<'\r'<<'\n'); procedure with constant arguments
```

Notes regarding functions and procedures

Observe the following notes.

Note

Double inverted commas

If user interface text contains a function in double inverted commas `"`, change the double inverted commas into single inverted commas `' " '`, so that these double inverted commas are not interpreted by the CMC.

Example:

```
Msg("Component '"' HMI Install '"' ...")
```

This procedure must be used for all double inverted commas within strings, as is typically also the case in the NC.

Note

Incorrect formatting

If an error occurs when formatting texts, this can lead to the entire formatting of the logbook being changed or corrupted starting at this error. This is the case, for example, if a tag is not closed or is written incorrectly.

Note

Line breaks and text formatting

Line breaks can be inserted in the user interface texts via `'<br>'`. Texts can be formatted.

You can find more information in Section "Text formatting options (Page 90)".

Note

Regular expression as argument

If a regular expression is used for an argument of a function or a procedure, please note that you should take considerable care when using metacharacters as character literals.

You can find more information at Create MyConfig - Diff in Section "Using regular expressions in Diff (Page 206)".

6.13 Commenting

Commenting - `; Comment`

Comments are identified by a semicolon as prefix. As a consequence, all of the characters up to the end of the line are no longer evaluated.

Example

```
;Comment across complete line
N20000 $MC_CHAN_NAME="Machine" ;Comment in line
```

6.14 Area data

Area data - `$MN_?, ... , $MD_?, p?, $MM_?, ...`

All variables/identifiers without the package prefix "`Up.`" are interpreted as area data. This includes, for example, machine data, setting data, option data, safety data, tool data, drive data and user data.

The area data can be written on the **left side** of an assignment and be read on the **right side** of an assignment.

You insert area data in the "Script Editor":

1. Right-click to open the "Insert identifier" dialog via the shortcut menu commands "Insert script" > "Area data...".
2. Select the desired identifiers by double-clicking them.

Furthermore, you can insert the area data from Diff via the Windows clipboard.

Note

You can obtain the options for cross-area reading and writing area data from Section "Data accesses (Page 50)".

6.15 Data accesses

Alternative data accesses

As an alternative to setting a "Section (Page 53)", you can also specify the section directly when accessing using a fully qualified identifier.

Specifying the section has a higher priority when compared to the last setting for the section.

Section setting: `[C?],[B?_S?_PS??]`

Read access to NC, DRV and HMI data from dialog scripts is only possible via the fully qualified identifiers.

Overview of data accesses

The following is shown in the following table:

- At which locations of the script processing you can read or write data with which syntax
- Data and their associated notation are listed as examples.

Table 6-5 Overview of data accesses: Read only/read and write

Data accesses	Dialog pages	Step tree			
	Cross-area		Area specific		
	Dialog scripts	Action scripts	Scripts		
			All	NC data	DRV data
Read only	Fully qualified identifiers for Area data NC, DRV, HMI				
	`Up.ChanNameC1 = NC[C1].$MC_CHAN_NAME` `Up.P105Value = PS[B3_S3_PS3].P105`				

Data accesses	Dialog pages	Step tree			
		Cross-area		Area specific	
	Dialog scripts	Action scripts	Scripts		
			All	NC data	DRV data
Reading and writing	Package variables – "Up.name"				
	Only package variables – "Up.name"		Simple identifiers with fully qualified section	Simple identifiers with simple section Simple identifiers with fully qualified section	

	`Up.Var1`		`Up.Var1`
			
			`NC[C1]`
			`$MC_CHAN_NAME`
			
			`PS[B3_S3_PS3]`
			`P105`
				`Up.Var1`
				
				`[C1]`
				`$MC_CHAN_NAME`
				
				`Chandata(1)`
				`$MC_CHAN_NAME`
				
				`NC[C1]`
				`$MC_CHAN_NAME`
					`Up.Var1`
					
					`[B3_S3_PS3]`
					`P105`
					
					`PS[B3_S3_PS3]`
					`P105`

Syntax for fully qualified identifiers

The syntax for a fully qualified identifier is generally:

```
<Namespace>[<Section>].<Variable>
```

Example:

```
NC[C1].$MC_CHAN_NAME
```

Syntax for simple identifiers

The syntax for simple identifiers with a simple section is generally:

- [Section]
  Variable
- CHANDATA (channel)
  Variable

Below a section specification any number of variables/identifiers and scripts belonging to this section can be used for assignments or queries.

Syntax for simple identifiers with a fully qualified section

The syntax for simple identifiers with a fully qualified section is generally:

- Namespace[section]
  Variable

Below a fully qualified section, any number of variables/identifiers and scripts belonging to this section can also be used for assignments or queries.

Namespaces and sections

The following namespaces and sections are defined:

Table 6-6 Namespaces and sections

Target area	Namespace	Section	Example

NC data	NC	Channel C1, C2 ...C10	`NC[C2]$MC_CHAN_NAME="Test"`
			
			`If NC[C1].$MC_CHAN_NAME==...`

SINAMICS data	PS	**B<x>_S<y>_PS<z>**
		Bus no. for x
		Slave no. for y
		DO no. for z
			`PS[B3_S3_PS1]p105=1`
			
			`If PS[B3_S3_PS1].p105==0`

Example: Cross-area read-access in script

```
If NC[C1].$MC_CHAN_NAME != "Test"
```

Example: Cross-area access in script (NC)

```
[C1] ; Global channel setting, as an alternative to CHANDATA(1)


; Test whether channel name is missing for channel 1 and is available
for channel 2
If $MC_CHAN_NAME == null && NC[C2].$MC_CHAN_NAME != null


; Copy channel name from channel 2 to channel 1
 $MC_CHAN_NAME = NC[C2].$MC_CHAN_NAME


EndIf
```

Example: Set channel

Examples

```
[C1]
;channel 1 is set as an area from here
$MC_CHAN_NAME = "Channel 1"
Up.NAME_CHAN_3 = NC[C3].$MC_CHAN_NAME ; access across areas
to channel 3
Up.NAME_CHAN_1 = $MC_CHAN_NAME ; access to channel 1
[C2]
;channel 2 is set from here
$MC_CHAN_NAME = "Channel 2"
CHANDATA(3) ;alternative NC-adequate long notation
;channel 3 is set from here
```

6.15.1 Sections

Within the target areas (NC data, SINAMICS data), the structured data is addressed by specifying a section. Set sections act local to a script until another section is set or the script (task) ends.

Further expansions can be found in Section "Data accesses (Page 50)".

Set section - `[B?_S?_PS?], [C?], CHANDATA(?)`

For data with the same name, which occur in various areas, before assigning a value, a section must be specified. This section is effective as long as no other section is specified. Data accesses between areas are unaffected by this.

When no channel is specified for NC data, then channel "1" is assumed.

The following section data is permissible depending on the target area:

Target area	Explanation	Format
SINAMICS data	Subpath within the drive system	`[B3_S3_PS3]`

NC data channel assignment within the NC [C1] to [C10] or CHANDATA (1) to CHANDATA (10) for NC channels 1 to 10.

Examples

```
[B3_S3_PS3]                    ; SINAMICS drive data
p1082[1] = 9000                ; Maximum speed
```

Drive data

Syntax	Description
`PS[B<bNr>_S<sNr>_PS<psNr>].<drvData>`	Value of the SINAMICS drive parameter "drvData" of the specified section: • Bus with the bus number "bNr" • Slave with the slave number "sNr" • Drive object with the number "psNr" Values from 1 to 99 are valid for "psNr". **Example**: `PS[B3_S3_PS1].p105`

6.16 Control structures

6.16.1 Control structures - context menu

Script Editor

In the "Script Editor" integrated in Expert, the "Insert script" shortcut menu provides support when creating a control structure.

Control structures used in CMC

The following variants of the control structures are available:

- If control structure (Page 55)
- If-Else control structure (Page 55)
- If-ElsIf control structure (Page 55)
- If-ElsIf-Else control structure (Page 55)
- While control structure (Page 56)

Display and use only in the preprocessor script:

- #Pragma Eol control structure (Page 57)
- #Rem control structure (Page 57)

6.16.2 If control structure / If-Else control structure

Conditional execution - `If ... Else ... EndIf`

```
If exp
    block1
Else
    block2
EndIf
```

The control structure allows the conditional execution of two different instruction blocks depending on whether the "exp" condition is fulfilled or not. If it is fulfilled (true), then the first operation block "block1" is executed, else the second execution "block2" is executed.

Each of the two instruction blocks can, in turn, contain control structures; this means that any nesting depth is possible.

Example

```
CHANDATA(1)
If Up.Velo_X != null
  If Up.Velo_X > ($MA_MAX_AX_VELO[AX1]+ 100)
    $MA_MAX_AX_VELO[AX1] = Up.Velo_X
  Else
    $MA_MAX_AX_VELO[AX1] = 1000
  EndIf
EndIf
```

6.16.3 If-ElsIf control structure / If-ElsIf-Else control structure

After the If condition alternative conditions can be checked by specifying ElsIf or ElIf.

The instruction block is executed after the first fulfilled condition. The instruction block after Else is executed if no condition is fulfilled.

Conditional execution - If ... ElsIf... Else... EndIf

```
If exp
 block_1
ElsIf exp_2
 block_2
ElsIf exp_3
 block_3
…
Else
 block_E
EndIf
```

Note

For the alternative conditions, the keywords "`ElsIf`" and "`ElIf`" are used as equivalents.

The control structure allows the conditional execution of different instruction blocks depending on whether the "exp" condition is fulfilled or not.

If this is fulfilled (true), then the first instruction block "block_1" is executed, otherwise it is checked whether the second alternative condition "exp_2" is fulfilled or not.

If this is fulfilled (true), then the second instruction block "block_2" is executed, otherwise it is checked whether the next alternative condition "exp_3" is fulfilled or not.

If this is fulfilled (true), then the third instruction block "block_3" is executed, and so on.

Once a condition or alternative condition is fulfilled, the associated instruction block is executed and then the execution continues after the associated keyword "EndIf".

As long as the condition or alternative condition is not fulfilled, the subsequent alternative condition is checked.

If no alternative condition is fulfilled, the instruction block "block_E" in the alternative branch is executed after the keyword "Else".

If no alternative branch "Else" was specified, then processing is continued after the keyword "EndIf" without executing an instruction block.

Each instruction block can, in turn, contain control structures; any nesting depth is possible.

Example 1

```
CHANDATA(1)
If Up.Velo_X != null
  If Up.Velo_X > $MA_MAX_AX_VELO[AX1]
    $MA_MAX_AX_VELO[AX1] = Up.Velo_X
  Else
    $MA_MAX_AX_VELO[AX1] = 1000
  EndIf
EndIf
```

Example 2 - Alternative condition

```
IF (up.selection == 1)
 Log("selection 1")
ELSIF (up.selection == 2)
 Log("selection 2")
ELIF (up.selection == 3)
 Log("selection 3")
ELSE
 Log("selection was not 1, 2 or 3")
ENDIF
```

6.16.4 While control structure

Repeated execution - `While ... EndWhile`

```
While exp
     block
EndWhile
```

The control structure allows the repeated execution of the operation block "block" as long as the "exp" condition is fulfilled.

The operation block can, in turn, contain control structures; this means that any nesting depth is possible.

Example

```
[B3_S3_PS99]
Up.i = 0
Up.CU_Hat_Servo = false
While Up.i < 6 && !Up.CU_Hat_Servo
  If p978[$(Up.i)] != 255
    Up.CU_Hat_Servo = true
  EndIf
  Up.i += 1
EndWhile
```

6.16.5 Pragma Eol control structure

Examples for using the "`#Pragma Eol control structure`" are provided in Expert under "Actions in the script editor" > "Modify" > "Description of the Change action" > "examples of Change action with "Preprocessor script".

6.16.6 Rem control structure

An example for using the "`Rem control structure`" is provided in Expert under "Actions in the script editor" > "Modify" > "Description of the Change action" > "Examples of change action with "Preprocessor script".

6.17 Operators

6.17.1 Overview operators

The following operator groups are available for scripts:

- Assignment operators (Page 58)
- Arithmetic operators (Page 59)
- Logic operators (Page 60)
- Comparison operators (Page 60)
- Special operators (Page 61)
- String concatenation (Page 62)

6.17.2 Assignment operators

Assignment operators - `VAR = 123`

The assignment operators constitute the core function of scripts, as they write the control data. All assignment operators can also be applied to script variables.

Table 6-7 The following assignment operators are available:

?=	Assign if blank or not available Assign the specified value if no value is available. No assignment is made if a value is already available.
=	Assign Assignment of a new value. The assignment is always executed. If the data item or variable does not yet exist, then it is created. If the data item or variable already exists, then the value is changed. An assignment of "null" deletes the variables or resets the value to the default value.
~	Delete The data and the value are removed. For control data (NC/drive data), this generally means that the value is reset to the default value of the control (of the software release).
+=	Add Addition of the specified value to the existing value.
-=	Subtract Subtraction of the specified value from the existing value
*=	Multiply Multiplication of the specified value and the existing value.
/=	Divide Division of the existing value by the specified value.
|=	OR operation OR operation on the specified and existing value (set bits).

&=	AND operation AND operation on the specified and existing value (reset bits).
:=	Assignment without evaluation Assignment without evaluation to the right of the operator. The value is transferred to the control as specified. Everything to the right of "=" (e.g. a variable) is neither interpreted by the package nor processed.

An error message is output if a value does not exist for compounded operators "`+= -= *= /= &= |=`".

Examples

```
N20000 $MC_CHAN_NAME="TUTORIAL"
$ON_AXIS_FUNCTION_MASK~
$MA_AX_VELO_LIMIT[0,AX1]?=2300
R[0]+=10
$MN_MM_TOOL_MANAGEMENT_MASK|='B10000'
GUD_2=$MA_MAX_AX_VELO[AX1]
Up.myVar ~ ; Delete variable
Up.myVar = null ; Delete variable (alternative)
```

6.17.3 Arithmetic operators

Arithmetic operators - `VAR = 1 + 2 - 3`

Arithmetic operators permit the assignment of a term comprising several constants and variables.

The following arithmetic operations are supported:

Table 6-8 Arithmetic operators

+	Add
-	Subtract
*	Multiply
/	Divide
|	OR operation
&	AND operation

The calculation is made for + - * / in double format (15-digit accuracy) and for & | in UInt64 format.

Calculations must be set in brackets in order to define an unambiguous sequence.

Examples

```
UP.Index=UP.Index +1
$MA_MAX_AX_VELO[AX1]= $MA_MAX_AX_VELO[AX2]*1.3
GUD_1=(100-50)/3
$MN_COMPAR_TYPE_1 = $MN_COMPAR_TYPE_1 | 'B10000'
```

6.17.4 Logic operators / comparison operators

Condition- `true != false`

The following logical operations and comparison operators are available to form Boolean expressions (true/false):

Table 6-9 Logic operators

&&	Logic AND operation
||	Logic OR operation
!	Negation operator

Table 6-10 Comparison operators

==	Equal to
!=	Not equal to
<	Smaller
>	Larger
<=	Less than or equal to
>=	Greater than or equal to

Logical connections must be set in brackets in order to define an unambiguous sequence.

Example

```
if UP.StepActive && ((Version(PCU, "SINUMERIK Operate") <=4.6) || (Version(NCU, "InternalVersion") ==2.7))
  Msg("The version of PCU or NCU is not valid for this step.")
endif
```

Note

Use of functions permitted

You can also use functions in logical operations.

Example

```
if Input(("Number of magazine locations")>60)
```

Note

Calculation in logical operations

Calculation is also possible in logical operations.

Example

```
If ((Up.Velo + Up.Offset)>2000)
```

6.17.5 Special operators

Replacement operator - $(up.?)

The "replacement operator" makes it is possible to replace parts of an instruction, an identifier, a value or a message text flexibly with the value of a package variable "Up.identifier".

Note the following rules when applying the "replacement operator":

- Use only with package variables.
  More information is provided in the section "Package variables (Page 44)".
- During replacement, straight quotation marks are removed from string values.
- "Replacement operators" can be nested.

Examples for applying the "replacement operator":

- In sections
  `[$(up.doX.pspath)]` or `[B3_S$(Up.slave)_PS$(Up.doNr)]`
- In identifiers
  `$MA_MAX_AX_VELO[AX$(UP.AXNr)]=1000`
- In parameter indices
  `If p107[$(Up.i)] == 254`
- In comparison operations
  `Version(PCU, "SINUMERIK Operate")>=$(Up.HMI)`
- In functions and procedures
  `MSG("Maschinennummer: $(Up.Masch_Nr)")`

Note

Instruction lines in which a "replacement operator" is used are not checked for correct syntax in a check run.

Note

The "replacement operator" is not used in the property fields of the configuration (configuration, dialogs and steps).

6.17.6 String concatenation

Using the string concatenation operator "<<" you can combine constant strings and variables to form a resulting STRING.

The string concatenation operator is primarily intended for assigning "Up variables" for their use in functions or procedures with string parameters.

The string concatenation operator can also be directly used in functions and procedures.

Example

```
Up.real = 12.34
Up.text = "units"


Up.para = "Value of Up.real: " << Up.real << " " << Up.text
Log(Up.para)
```

6.18 String operations

6.18.1 StringLen

StringLen("<string>")

The `StringLen()` function determines the length of a `<string>` character string and returns it as a decimal number.

In principle, this function can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts and "CMC scripts in Diff (Page 151)".

Note

Counting characters

All characters of a character string are counted, the printable and also the non-printable.

`'\r'` and `'\n'` each count as one character and are counted.

Example

```
Up.Len = StringLen("Ingenuity for life") ; Up.Len = 18
```

6.18.2 StringMatch

StringMatch("<string>", "<search>")

The `StringMatch()` function searches in the `<string>` character string for the `<search>` search pattern.

In principle, this function can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts and "CMC scripts in Diff (Page 151)".

Regular expressions are used as search pattern. The return value of the function is a character string.

Note

Using metacharacters as character literals

Take special care when using metacharacters as character literals in a regular expression.

You can find more information on regular expressions at Create MyConfig - Diff, Section "Use of regular expressions in Diff (Page 206)".

Example

```
Up.NCU = StringMatch("NCU1750", "NCU([0-9]+)") ;Up.NCU "1750"
Up.NCU = StringMatch(Up.$Env.NCU, "NCU([0-9]+)") ;Up.NCU "1750"
```

6.18.3 StringPos

StringPos("<string>", "<search>", <pos>)

The `StringPos()` function searches in the `<string>` character string for the `<search>` character string, starting at the `<pos>` position.

In principle, this function can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts and "CMC scripts in Diff (Page 151)".

The `<search>` character string to be sought is specified via a regular expression. The case-sensitivity in the `<search>` argument is not relevant and is not observed.

The function returns the position of the first occurrence of `<search>` as decimal number. If the `<search>` character string was not found, a `-1` is returned.

When assessing the returned positions, note that the first character in the <string> is on the 0 position. Spaces in the <string> are also characters, non-printable characters.

Note
Using metacharacters as character literals

Take special care when using metacharacters as character literals in a regular expression.

You can find more information on regular expressions at Create MyConfig - Diff, in Section "Using regular expressions in Diff (Page 206)".

Example

```
Up.Pos=StringPos("Ingenuity for life", "e", 0)      ;Up.Pos = 3
Up.Pos=StringPos("Ingenuity for life", "E", 5)      ;Up.Pos = 17
Up.Pos=StringPos("Ingenuity for life", "e", 25)     ;Up.Pos = -1
Up.Pos=StringPos("Ingenuity for life", "x", 0)      ;Up.Pos = -1
Up.Pos=StringPos("Ingenuity for life", "for", 0)    ;Up.Pos = 10
Up.Pos=StringPos("Ingenuity for life", "[efg]", 0)  ;Up.Pos = 2
```

6.18.4 StringReplace

StringReplace("<string>", "<search>", "<replace>")

The StringReplace() function searches in the <string> character string for the <search> character string and replaces it with the <replace> character string.

In principle, this function can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts and "CMC scripts in Diff (Page 151)".

The return value of the function is a character string with the replaced characters.

The <search> character string to be sought is specified via a regular expression.

The case-sensitivity in the <search> argument is relevant and is observed. Replacement is made only when identical. If nothing to be replaced is found, the function returns the <string> argument unchanged.

Multiple occurrences of the sought <search> characters are all replaced.

Note
Using metacharacters as character literals

Take special care when using metacharacters as character literals in a regular expression.

You can find more information on regular expressions at Create MyConfig - Diff, Section "Use of regular expressions in Diff (Page 206)".

Example

Use the name of the input archive for the name of the package execution logbook.

```
Up.$Dialog.PackageEnd.LogName =
StringReplace(Up.$Dialog.ArcSelection.ArchiveIn, ".arc","")
```

The use of an "NC.ARC" input archive by this instruction produces the "NC.htm" logbook. As the file extension is replaced with an empty string and the package automatically adds the "*.htm" extension when the logbook is saved.

6.18.5 StringSubStr

StringSubStr("<string>", <pos>, <len>)

The StringSubStr() function returns a subcharacter string from the <string> character string, which starts at the <pos> position and has the length <len>.

In principle, this function can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts and "CMC scripts in Diff (Page 151)".

The <pos> and <len> arguments are of the decimal number type.

Note that the first character in the <string> is on the 0 position. Spaces in the <string> are also characters, non-printable characters.

If the subcharacter string was not found, a "" empty string is returned.

Example

```
Up.Str=StringSubStr("Ingenuity for life", 0, 9)     ; Up.Str =
                                                    "Ingenuity"
Up.Str=StringSubStr("Ingenuity for life", 10, 3)    ; Up.Str = "for"
Up.Str=StringSubStr("Ingenuity for life", 30, 9)    ; Up.Str = ""
Up.Str=StringSubStr("Ingenuity for life", 0, 0)     ; Up.Str = ""

If Up.Str == ""
  MSG("No subcharacter strings found!")
EndIf
```

6.19 File operations

The functions and procedures that you can use for files are described in the following sections.

In the subsequent sections, the following values are uniformly permissible for argument <area>:

- RTS
  - For the runtime system (NCU, PC/IPC) on which the package is executed
  - You can specify the path, either in relative terms, starting next to the package, or in absolute terms
  - The use of environment variables is possible
- ARC
  - For the archive folder, i.e. the image of the SINUMERIK archive
  - You specify the path in relative terms, starting with NC, PLC, or DRV
- NCU
  - For the file system of the NCU
  - You specify the path in absolute terms, starting with "/" and taking into account uppercase/lowercase (case sensitive)
- PCU
  - For the file system of the PCU
  - You specify the path in absolute terms, starting with the drive letter, e.g. "D:\" or "D$\"

Note

Activating data areas

Make sure that you only reach the areas that you have activated in Expert in the "Project explorer" in the "Configuration (Page 236)" under the "Data areas" property group.

6.19.1 FileCopy

FileCopy(<area>, "<sourcepath>", <area>, "<destinationpath>")

The FileCopy() procedure copies a file from a source to a target across areas.

In principle, this procedure can be called in all scripts, "dialog scripts (Page 242)", "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

The areas for the source and target are specified via the <area> argument.

The permissible values for the <area> argument are provided in Section "File operations (Page 65)".

The <sourcepath> argument specifies the source path of the file to be copied.

The <destinationpath> argument specifies the target path of the file.

The paths always contain the storage location and the file name. If the file to be copied already exists in the target, it will be overwritten **without** a prompt.

If the package runs on the NCU, copying to the PCU via <area> PCU is not possible.

Example 1

```
FileCopy(RTS, ".\Test.txt", NCU, "/user/Test_NCU.txt")
```

Example 2

Version files of the NCU are stored in the CF card folder next to the package.

```
Up.cf = "./CF-" << Up.$Env.CFID << "/"
FileCopy(NCU, "/user/system/etc/hwversions.xml", RTS, "$
(Up.cf)hwversions.xml")
FileCopy(NCU, "/siemens/versions.xml", RTS, "$
(Up.cf)siemens_versions.xml")
```

6.19.2 FileDelete

FileDelete(<area>, "<path>")

The `FileDelete()` procedure deletes a file in a target area.

In principle, this procedure can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts, and "CMC scripts in Diff (Page 151)".

The target area is specified using the <area> argument, in which the file to be deleted is located.

The permissible values for the <area> argument are provided in Section "File operations (Page 65)".

The <path> argument specifies the path of the file to be deleted and the file name.

If the package runs on the NCU, deletion on the PCU via <area> PCU is not possible.

Example of FileDelete

```
FileDelete(RTS, ".\Test.txt")
```

6.19.3 FileExist

FileExist(<area>, "<path>")

The `FileExist()` function checks whether a folder or a file exists.

In principle, this function can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts and "CMC scripts in Diff (Page 151)".

If the folder or the file with the absolute path <path> exists in the area <area>, the FileExist() function returns the value "true".

Absolute and relative paths are permitted for the <path> argument. Relative paths refer to the package location.

The use of environment system variables and package variables is possible and permitted.

The permissible values for the <area> argument are provided in Section "File operations (Page 65)".

Note

Observe case sensitivity

Please observe the case sensitivity in the path on the NCU/PPU under Linux.

Make sure that you only reach the "Data areas" that you have activated in Expert in the "Project explorer" in the "Configuration (Page 236)" under the "Data areas" property group.

Call examples

```
FileExist(NCU, "/siemens/versions.xml")
```
;searches for the file on the NCU.

```
FileExist(NCU, "/install/arc")
```
;searches for the folder on the SD card of the NCU.

```
FileExist(PCU, "C:\Users")
```
;searches for the folder in drive C of the PCU.

```
FileExist(PCU, "D:\UP")
```
;searches for the folder on drive D of the PCU.

```
FileExist(ARC, "NC\CST.DIR\TEST.SPF")
```
;searches for the file in the archive(image).

```
FileExist(RTS,"MyConfig.upcfg")
```
;searches for the file next to the package.

```
FileExist(RTS,"%UP_PACK%/MyConfig.upcfg")
```
;path specification with "environment system variables" under Windows, searches for the file next to the package.

```
FileExist(RTS,"$(UP_PACK)/MyConfig.upcfg")
```
;path specification with "environment system variables" under Linux, searches for the file next to the package.

```
FileExist(RTS,"$(Up.Pfad)/MyConfig.upcfg")
```
;path specification via script variable.

```
FileExist(RTS,"$(Up.$Pack.Dir)/MyConfig.upcfg")
```
;path specification via system variable.

Example of the output of a message

Output of a message if the "/siemens/versions.xml" file was found on the NCU/PPU.

```
IF FileExist(NCU, "/siemens/versions.xml")
  Msg("File 'versions.xml' is available on the NCU.")
ENDIF
```

Example of package variable `UP.DH_exists`

If data management "`F:\DH`" was found on the PCU, then package variable "`UP.DH_exists`" has the value "true".

```
Up.DH_exists = FileEXIST(PCU, "F:\DH")
```

6.19.4 FileRead

FileRead(<area>, "<path>")

The `FileRead()` function reads the contents of a text file `<path>` from the `<area>` area and returns it as a character string.

In principle, this function can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts and "CMC scripts in Diff (Page 151)".

For the `<path>` argument, absolute paths are permitted in the NCU/PCU/RTS areas and relative paths in the ARC/RTS areas.

The use of "environment system variables" and package variables is possible and permitted.

The permissible values for the `<area>` argument are provided in Section "File operations (Page 65)".

Examples

```
Up.File = FileRead(NCU, "/siemens/versions.xml")
```

```
Up.File = FileRead(ARC, "NC\CST.DIR\Test.SPF")
```

`Up.File = FileRead(RTS,"MyText.txt")`; reads the file next to the package.

`Up.File = FileRead(RTS,"%UP_PACK%/MyText.txt")` ;path specification with "environment system variables" under Windows, reads the file next to the package.

`Up.File = FileRead(RTS,"$UP_PACK/MyText.txt")` ;path specification with "environment system variables" under Linux; reads the file next to the package.

`Up.File = FileRead(RTS,"$(Up.Pfad)/MyText.txt")` ;path specification via script variable.

`Up.File = FileRead(RTS,"$(Up.$Pack.Dir)/MyText.txt")` ;path specification via system variable.

Note

Determination of the end of line / line break

With the `StringPos()` function, you determine which end of line / line break characters were used in the file.

\r → Carriage return (CR or 0x0D)

\n → Line feed (LF or 0x0A)

Example

```
Up.File = FileRead(RTS,"MyText.txt")
If StringPos(Up.File, "\r\n", 0) != -1
  MSG("It is a DOS/Windows file.")
ElsIf StringPos(Up.File, "\n", 0) != -1
  MSG("It is a Linux (UNIX) file.")
ElsIf StringPos(Up.File, "\r", 0) != -1
  MSG("It is a Mac file.")
Else
  MSG("The content of the file comprises only one.") line
EndIf
```

Note

Observe case sensitivity

Observe case sensitivity in the path on the NCU/PPU under Linux.

Make sure that you only reach the "Data areas" that you have activated in Expert in the "Project explorer" in the "Configuration (Page 236)" under the "Data areas" property group.

6.19.5 FileWrite

FileWrite(<area>, "<path>",“<string>”)

The `FileWrite()` procedure writes the character string `<string>` to a text file `<path>` in the `<area>` area.

In principle, this procedure can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts, and "CMC scripts in Diff (Page 151)".

The procedure always creates a new file.

Using the <path> argument you specify the file with path and name in which the character string is to be written.

You can specify the path <path> either relative to the package - or in absolute terms.

The use of environment variables is possible. Observe the case sensitivity under Linux.

The <string> character string may include characters of the UTF-8 character set or non-printable characters.

The permissible values for the <area> argument are provided in Section "File operations (Page 65)".

Examples

```
FileWrite(NCU, "/user/myfile.xml", Up.myfile).
```

```
FileWrite(ARC, "NC\CST.DIR\Test.SPF", Up.myfile)
```

```
FileWrite(RTS,"MyText.txt", Up.myfile)
```
;creates the file next to the package.

```
FileWrite(RTS,"%UP_PACK%/MyText.txt", Up.myfile)
```
;path specification with environment system variables under Windows; creates the file next to the package.

```
FileWrite(RTS,"$UP_PACK/MyText.txt", Up.myfile)
```
;path specification with environment system variables under Linux; creates the file next to the package.

```
FileWrite(RTS,"$(Up.Pfad)/MyText.txt", Up.myfile)
```
;path specification via script variable.

```
FileWrite(RTS,"$(Up.$Pack.Dir)/MyText.txt", Up.myfile)
```
;path specification via system variable.

Note

Use of the end of line / line break

If your character string has several lines, you can implement it with the string concatenation operator.

You can find more information in Section "String concatenation operator (Page 62)".

Example

```
FileWrite(RTS,"MyText.txt","Zeile 1"<<'\r'<<'\n'<<"Zeile
2"<<'\r'<<'\n')
```

\r → Carriage return (CR or 0x0D)

\n → Line feed (LF or 0x0A)

Note

Attaching characters to an existing file

If you attach characters to an existing file, you can implement this in two ways:

1. Use the `TraceToFile()` procedure. This procedure has the advantage that it always has high performance.
   Only use `TraceToFile()` for the runtime system (RTS). First copy the file to the runtime system, attach the file and then copy back the file.
   This procedure can be used when you write several times to the file and only copy the file back after all the write operations.
   You can find more information in Section "TraceToFile (Page 76)".
   Example
   ;If the package is executed on the PG/PC
   ```
   FileCopy(PCU, "D:\LastUpdate.txt", RTS, "LastUpdate.txt")
   TraceToFile("LastUpdate.txt", "Step 4711 successful")
   ...
   FileCopy(RTS, "LastUpdate.txt", PCU, "D:\LastUpdate.txt")
   ```
2. Read the contents of a file with the `FileRead()` procedure.
   Add your characters with the string concatenation operator (Concate-Operator) "<<" and write the STRING back to the file with `FileWrite()`.
   Example
   ```
   FileWrite(RTS, "RW.txt",FileRead(RTS,"RW.txt")<<'\r'<<'\n'<< "my attached line")
   ```

6.19.6 QueryIni

QueryIni("<path>", "<section/id>")

The `QueryIni()` function reads values from INI files.

In principle, this function can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts and "CMC scripts in Diff (Page 151)".

You can use this function for all files whose structure corresponds to the format of an INI file and are located on the runtime system (RTS) of the package execution.

These files have the following structure:

```
[Section]

Identifier = value
```

Using the `<path>` argument, you specify the file (with path and file name) from where a value is read.

You can specify the file path fully qualified in accordance with the runtime system - or also relative to the package.

Using `<section/id>`, you specify the section and the identifier for the value to be read. `<section/id>` is not case sensitive.

Exception: `<path>` on the NCU (Linux) is case sensitive.

The return value of the function is the value of the specified identifier in the section of the file.

The function returns "null" if the specified file, the section or the Identifier is not found. Or, if there is no equal character and no value at the identifier.

If there is an equals sign, but no value at the queried Identifier, the function returns an empty string "".

You can use the environment system variables of the runtime system.

By testing the return value for "null", you can determine whether or not the function could find the file, the section or the identifier.

Example 1

Query the "FirewallOpenPorts" identifier in the "LinuxBase" section in the "basesys.ini" file.

Content of "basesys.ini"

```
[LinuxBase]
FirewallOpenPorts="TCP/5900 TCP/102 TCP/22"
```

Function call in Expert

```
Up.FOP = QueryIni("/user/system/etc/basesys.ini","LinuxBase/
FirewallOpenPorts")
MSG("The FirewallOpenPorts are:$(Up.FOP)")
```

Function return value

Up.FOP contains the function return value: e.g. "TCP/5900 TCP/102 TCP/22".

Example 2

The softkeys are checked for their existence and access level in "logdrive.ini".

```
Up.file = "/user/sinumerik/hmi/cfg/logdrive.ini"
If !FileExist(RTS,Up.file)
 Warning("No 'logdrive.ini' file exists!")
Else
 up.i=1
 While up.i<25
  If QueryIni(Up.file,"DEVICE_$(up.i)/AccessLevel") == null
   Log("The $(up.i) softkey is not used.")
  Else
   up.AL=QueryIni(Up.file,"DEVICE_$(up.i)/AccessLevel")
   If up.AL != "7"
    Warning("The $(up.i) softkey has the AccessLevel $(up.AL)")
   Else
```

```
    Log("The access level of softkey $(up.i) is $(up.AL).")
   EndIf
  EndIf
  up.i+=1
 EndWhile
EndIf
```

6.19.7 QueryXml

QueryXml("<path>", "<xpath>")

The `QueryXml()` function reads the contents of XML files that are on the runtime system (RTS) of the package execution.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts, "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

The `<path>` argument specifies the XML file with path and file names (file path) from where the content of an XML node should be read. Whereby, the file path can be specified fully qualified, in accordance with the runtime system, or also relative to the package.

For Linux (NCU), the case sensitivity of the path is relevant, but not for Windows (PC/IPC).

The `XPath` expression in the `<xpath>` argument selects the XML node whose content should be output.

The case sensitivity for `<xpath>` is always relevant.

If the selected XML node was found by `<xpath>`, its content is returned as STRING. All other XML nodes are ignored.

If the XML file or the specified node is not found, then the function returns "null".

Note

XML node

An XML node can be, for example, an element, an attribute or text.

Note

XML node <xpath> return

If several XML nodes selected by `<xpath>` are found, only the content of the first XML node is returned.

Note

Use of environment variables

Environment variables of the runtime system can be used in the `<path>` argument.

Note

Return value "null"

By checking the return value for "null", it can be determined whether the function could not find the file or the XML node.

Note

Warning in the logbook if the file is not an XML file

If the specified file is not an XML file, the function behaves as if the specified path is invalid.

The function provides the return value "null" and writes the message of the invalid path as warning in the logbook.

Command line example

```
Up.valCustomerId = QueryXml( "/user/sinumerik/hmi/data/version/
identity.xml", "/Config_Identity/Addresses/Manufacturer/ Customer-
Id/text()")
```

Example

A specific alarm text from the "`alarmtxt_nculin_deu.ts`" input file should be found and its translation output as a file.

Extract from the input file "`alarmtxt_nculin_deu.ts`":

```
<TS>
  [...]
  <message>
    <source>510104/PLC/PMC</source>
    <translation>Axes not referenced, tool changer locked</
    translation>
  </message>
  [...]
</TS>
```

Content of the script:

```
;Initialization of the output file (text)
Up.pathToOutput = "Example2\Output.txt"

;If the output file exists on the RTS, it will be deleted
if FileExist(RTS,Up.pathToOutput)
 FileDelete(RTS,Up.pathToOutput)
endif
```

```
;Initialization of the input file (XML)
Up.pathToInput = "/user/sinumerik/hmi/lng/alarmtxt_nculin_deu.ts"

;Initialization of the Alarmtext ID
Up.alarmTextId = "510104/PLC/PMC"

;Initialization of the XPath expression with integration of the
Alarmtext ID as string
Up.xpathAlarmtext = "/TS/message[source/text()='$(Up.alarmTextId)']/
translation/text()"

;Use of the XPath expression on the input file with QueryXml
Up.valAlarmtext = QueryXml( Up.pathToInput, Up.xpathAlarmtext)

;If the alarm text was found, it will be output
if Up.valAlarmtext != null
 TraceToFile(Up.pathToOutput,"Alarm text found: $(up.valAlarmtext)")
endif
```

Content of the output file `"Example2\Output.txt"`:

```
Alarm text found: Axes not referenced, tool changer locked
```

6.19.8 TraceToFile

TraceToFile("<path>", "<string>")

The `TraceToFile()` procedure adds a `<string>` character string to a `<path>` file.

In principle, this procedure can be called in all "dialog scripts (Page 242)", "action scripts (Page 356)", scripts, and "CMC scripts in Diff (Page 151)".

If the file does not exist, it is created by the procedure. If the file already exists, the text is added at the end of the file.

The `<path>` argument specifies the file with path and name.

The path specification `<path>` can be relative to the package or absolute, but always refers to the runtime system of the package execution.

The use of environment variables is possible. Observe the case sensitivity under Linux.

The text including a line feed to be written to the file via the `<string>` argument is added at the end of the file:

- \r\n = CRLF = 0D0A under Windows
- \n = LF = 0A under Linux

The text may include characters of the UTF-8 character set.

Note

Possible uses

The `TraceToFile()` procedure only works in the runtime system (RTS) of the package execution and therefore has a high performance. You can use the procedure, for example, to write information from the package execution in an uncomplicated way to a file.

Example 1 of TraceToFile

```
TraceToFile("D:\Text.txt", "This is only the first test.")
```

Example 2 of TraceToFile

The channel name and the name of the first geometry axis are written to an Excel table.

```
Up.file = "Test.csv"
Up.i = 1
If FileExist(RTS, Up.file)
 FileDelete(RTS, Up.file) ; New CSV should be created
EndIf
TraceToFile(Up.file, "Channel name; 1st geometry axis") ; header line
While Up.i < NC[C1].$ON_NUM_CHANNELS
 TraceToFile(Up.file, "" << NC[C$(Up.i)].$MC_CHAN_NAME << ";" << NC[C
 $(Up.i)].$MC_AXCONF_GEOAX_ NAME_TAB[0]); The first two "" generate
 an empty string
 Up.i += 1
EndWhile
```

6.20 Dialogs

You can call the functions and procedures subsequently listed in this section as dialogs on the user interface of the packet execution and in "CMC scripts in Diff".

The dialogs provide operators with information - or ask information from operators.

Note

Straight quotation marks

If user interface texts or string values in a function or procedure contain straight quotation marks `" "`, the straight quotation marks must be enclosed within single quotation marks `'"'`. In order that they are not interpreted by Create MyConfig, as is also usual in the NC.

Examples:

```
Msg("Component '"'HMI Install'"'...")
```

```
TraceToFile(Up.File,"Up.Text='"'my text'"'")
```

Note

Incorrect formatting

If an error occurs when formatting texts, e.g. because you did not close a "tag" or wrote it incorrectly, this can lead to the entire formatting of the logbook being altered or corrupted starting at this error location.

Note

Line breaks and text formatting

Line breaks can be inserted in the user interface texts via `'<br>'`.

You can format texts. You can find more information in Section "Text formatting options (Page 90)".

Note

Handling special characters in strings

The straight quotation mark " is enclosed by inverted commas '"' within a string, but only counts as one character.

You can only individually insert special characters for line feed '\n' and carriage return '\r' in character strings using a string concatenation operator (Page 62) <<. These characters also only count as one character.

6.20.1 Msg

Msg("<label>")

The procedure Msg() generates a message, which is confirmed using "OK".

In principle, this procedure can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

Any message text, which is also written in the logbook, represents the `<label>` argument.

Example

```
MSG("Hello World!")
```



Figure 6-3 Create MyConfig Shield - Msg

6.20.2 Warning

Warning("<label>")

The Warning() procedure generates a message with the prompt to either "Continue" or "Cancel" the package.

In principle, this procedure can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

Any message text, which is also written in the logbook, represents the `<label>` argument.

Example

```
Warning("File $(Up.File) was not found.")
```



Figure 6-4 Create MyConfig Shield - Warning

6.20.3 Error

Error("<label>")

The `Error()` procedure generates a message, which can only be confirmed with "Cancel package".

In principle, this procedure can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

Any message text, which is also written in the logbook, represents the `<label>` argument.

Example

```
Error("Things cannot continue like this!")
```

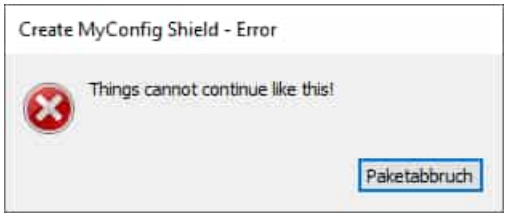


Figure 6-5 Create MyConfig Shield - Error

6.20.4 Introduction input functions

2 types of input functions are available:

- Input functions that are not linked to a particular type
- Input functions that are linked to a particular type

Input functions linked to a particular type return a typical value, contrary to input functions that are not linked to a particular type:

- "string"
- "integer"
- "real"
- Enumerator value as "string"
- "unsigned integer"

The type-specific input functions require a default value. The user-specified value can be specified directly or applied only at runtime via variables.

All of the input functions are always displayed in the "Step tree", independent of the "Display messages" selection.

Default value

The arguments of the type-specific input functions must include a default value so that fully automatic package execution is possible without requiring any operator action. You can change this default value when manually executing the package.

6.20.5 Input

Input("<label>")

The `Input()` function is a non-type-specific function.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

You query a value during script execution using the `Input()` function.

Whereby, the entered value determines the type. Only enter values that correspond to a permissible type, e.g. "1"; "5", "8"; "0x0"; "Test", true,....

If you do not enter any values, the function returns `"null"`. You can intercept this in the script code.

You explain the entry using argument `<label>`.

Example of Input

```
Up.in = null
While Up.in == null
 Up.in = Input("Make an entry.")
EndWhile
```

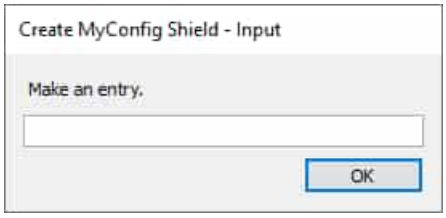


Figure 6-6 Example of Input

6.20.6 InputChoice

InputChoice("<label>", "<button1>;<button2>")

During package execution, the `InputChoice()` function displays a dialog with icons.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

In the dialog, you can make a direct selection by pressing individual buttons.

The selection that is made is returned as a character string (the button labeling).

The entry is explained using the `<label>` argument. Optionally, select an icon that is displayed in the dialog during packet execution in the argument `<label>`, at the beginning separated by a semicolon.

The following icons <label> are available:

- Selection without icon
- *I; selection with "Information" icon
- *Q; selection with "Question mark" icon
- *W; selection with "Warning" icon
- *E; selection with "Error" icon

The <button> argument is used to specify the identifiers for several buttons, separated by semicolons.

2 semicolons in a row cause a line feed, i.e. all of the subsequently listed buttons are displayed in a new line. In this way, buttons can also be arranged one below the another.

The freely selectable identifiers simultaneously define the possible return values of the function.

Control characters in the <button> argument:

- * Accept / input
- ^ Reject / cancel with ESC
- & Alt-Shortcut

The control characters '*' and '^' can be used to specify the default button for confirming or for cancelling with ESC at the beginning of an identifier.

Control character '&' in the identifier for buttons defines the subsequent letters as "Alt-shortcut" (shortcut Alt+<letter>).

During package execution, each button can also be operated via the keyboard, using the uniquely assigned keyboard shortcut. Ensure an unambiguous letter assignment across all buttons.

The "Alt-shortcuts" are not case sensitive.

Example, selection without icon

```
Up.ret = InputChoice("Example without icon", "*&OK")
```

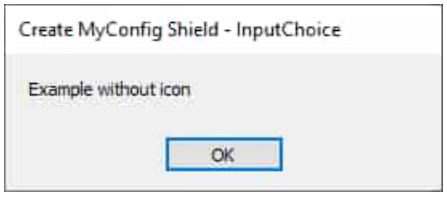


Figure 6-7 Example without icon

Example, selection with "Information" icon

```
Up.retI = InputChoice("*I; Example of icon '"'Information'"'",
"*&OK")
```



Figure 6-8 Example of an "Information" icon

Example, selection with "Question mark" icon

```
Up.retQ = InputChoice("*Q; Example of icon '"'Question'"'",
"*&Yes;&No")
```

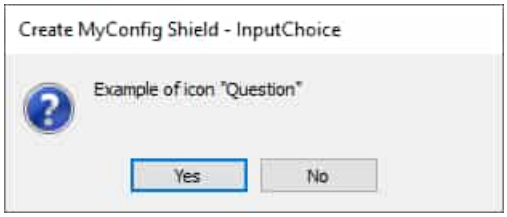


Figure 6-9 Example of a "Question" icon

Example, selection with "Warning" icon

```
Up.retW = InputChoice("*W; Example of icon '"'Warning'"'",
"*&OK;&Cancel")
```

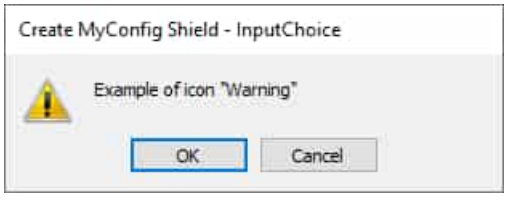


Figure 6-10 Example of a "Warning" icon

Example, selection with an "Error" icon

```
Up.retE = InputChoice("*E; Example of icon '"'Error'"'",
"*&Cancel;&Resume")
```

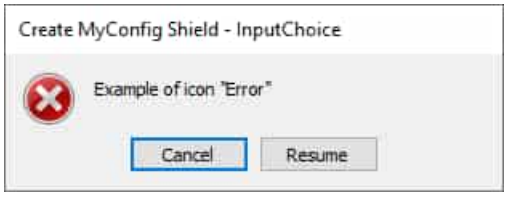


Figure 6-11 Example of an "Error" icon

Example for button operation

```
Up.example = InputChoice("*Q; Example of button operation with<br> *
Alt shortcut<br> * Input<br> * ESC", "*Alt+&A or Input; Alt+&B;^Alt
+&C or ESC")
```

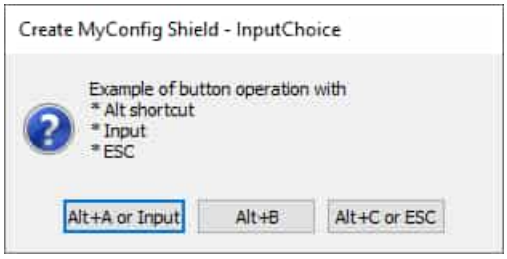


Figure 6-12 Example of button operation

Example of buttons one below the other

```
Up.MachineType = InputChoice("Choose the technology!", "&Milling;;
&Turning;; &Grinding")
```



Figure 6-13 Example of buttons displayed one below the other

6.20.7 InputEnum

InputEnum("<label>", "*<enum1>;<enum2>")

During package execution, function `InputEnum()` displays a dialog. You can make a selection in the dialog.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

The selected STRING is assigned to an Up variable.

- You can explain the entry using argument `<label>`.
- The selected text sequences can be defined using the `<enums n>` argument, each separated with a semicolon. Any number can be selected.
- You can define this as preselection with a "*" at the beginning of a selected text sequence. If you choose not to do this, then the first listed is preselected.
- The return value is the text that the user had selected as string.

Example with defined preselection (*)

```
Up.in = InputEnum("Please select input type:", "Magazine A;
*Magazine B; Magazine C")
```

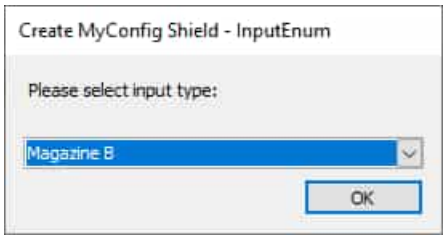


Figure 6-14 InputEnum - example with defined preselection

Example of preselection with replacement operator

Note the use of a replacement operator in the selection string.

```
Up.Type1 = "Milling machine"
Up.Type2 = "Drilling machine"
Up.Selection = "$(up.Type1); $(Up.Type2); turning machine"
Up.MachinenType = InputEnum("Which machine type?", "$(Up.Selection)")
```

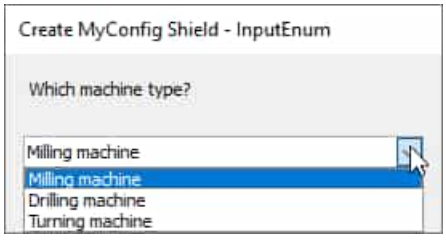


Figure 6-15 InputEnum - example preselection with replacement operator

6.20.8 InputInt

InputInt("<label>", <int>)

During package execution, function `InputInt()` displays a dialog in which an operator can enter an integer number.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

The integer value is assigned to an "Up variable".

- The entry is explained using argument <label>.
- The <int> argument must contain a default value. You can change the default value during package execution.
- The return value is the number that the user had entered.

Example of entering an integer

```
Up.Integer = InputInt("Enter an integer", 1234)
```

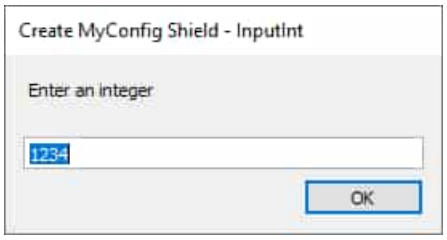


Figure 6-16 InputInt - example, entering an "integer number"

Example of replacement from STRING

For example, from an Enum

```
Up.n = "1500"
Up.Speed = InputInt("Confirm the speed.", $(Up.n))
```

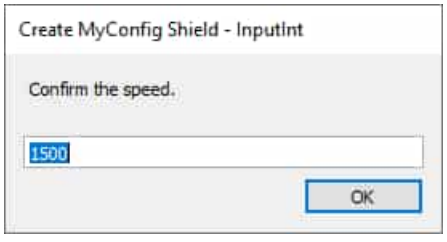


Figure 6-17 InputInt - example, "Replacement from STRING", e.g. Enum (enumeration)

6.20.9 InputReal

InputReal("<label>", <real>)

During package execution, function `InputReal()` displays a dialog in which an operator can enter a real number.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

The real value is assigned to an "Up variable".

- The entry is explained using argument `<label>`.
- The `<real>` argument must contain a default value. You can change the default value during package execution.
  The following entries are permitted:

Decimal:	-15	= -15
	1.5	= 1.5
Exponential:	1.5e2	= 150
	1.5ex+2	= 150
	-1.5ex-2	= -0.015

- The return value is the number that the user had entered.

Example of real-number entry

```
Up.RealNumber = InputReal("Enter a real number", 567.890)
```

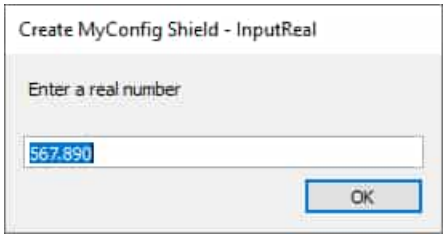


Figure 6-18 InputReal - Example of "Real number entry"

Example of specification with variables

```
Up.Exp = 12eX-3     ; = 0,012
Up.RealNumber = InputReal("Enter a number in exponential form:",
Up.Exp)
```

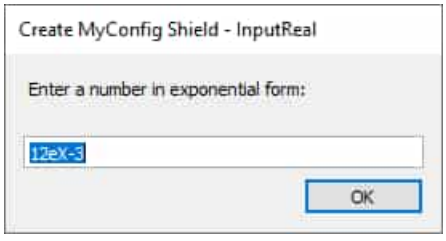


Figure 6-19 InputReal - Example of "Specification with variables"

6.20.10 InputText

InputText("<label>", "<string>")

During package execution, function `InputText()` displays a dialog. The user is prompted to enter a text.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

This text is assigned to an "Up variable".

- The entry is explained using the `<label>` argument.
- A text is preselected using the `<string>` argument.
- The dialog is switched to hidden password entry using the "*" user-specified value.
- The return value is the text that the user had entered.
- With an empty input field, pressing "OK" returns an empty string " ".

Example of InputText with and without replacement operators

```
Up.Name = InputText("Enter the machine name.", "'Machine XY'")
```

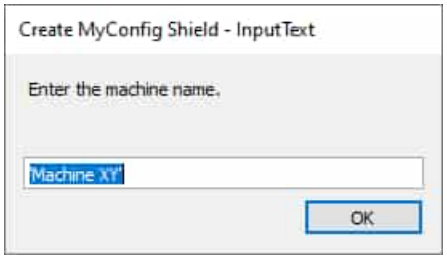


Figure 6-20 InputText - Example InputText without replacement operator

```
Up.FA = "Siemens"
Up.Name = "'Machine XY'"
Up.Manufacturer = InputText("From which company is $(up.Name)
manufactured?",up.FA)
```

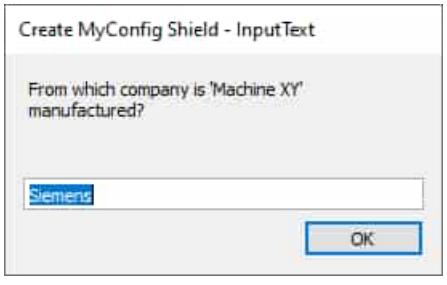


Figure 6-21 InputText - example of InputText with replacement operator

Example InputText for password entries

```
Up.Password = InputText("Enter your password", "*")
```



Figure 6-22 InputText - Use example InputText for entering passwords

6.20.11 InputUInt

InputUInt("<label>", <uint>)

During package execution, the `InputUInt()` function displays a dialog in which you must enter an unsigned integer number.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

The unsigned integer value is assigned to an "Up variable".

- You can enter the value in various formats:

```
   DEZ = 123
HEX_NC = 'H9F'
HEX_0x = 0x9F
BIN_NC = 'B1101'
  BiCo = 3:1.23
```

- The entry is explained using the `<label>` argument.
- The `<uint>` argument must contain a default value. You can change the default value during package execution.
- The return value is the number that the user had entered.

Example

```
Up.Uint = InputUInt("Enter an unsigned integer", 'H9F')
```

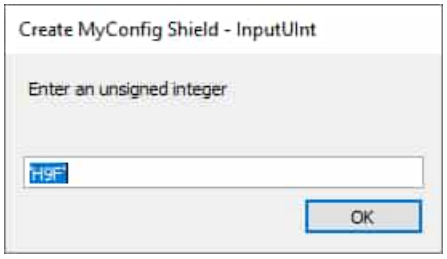


Figure 6-23 Example "Entering an unsigned integer number"

Example of specification with replacement operator

```
Up.Default = 0x9F
Up.Uint = InputUInt("Enter an unsigned integer", $(Up.Default))
```

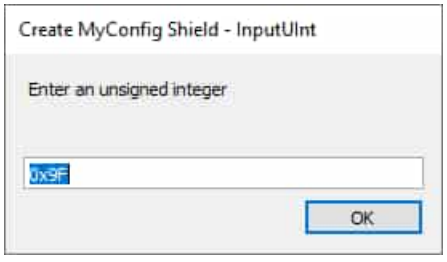


Figure 6-24 InputUInt - example "Specification with replacement operator"

6.20.12 Text formatting options

Programming options

Examples are listed below on how you can format texts.

A selection of the text formatting options is displayed when using the `InputReal` function. These formats are possible for all functions which use text and also for all other user texts in CMC, for example in notes relating to dialogs and steps.

Use HTML codes when formatting.

Example representation of a line break

For a clearer arrangement, insert a line break `<br>` between the header and the text.

```
up.wrap=InputReal("Notice! <br> Enter a number.",3.0)
```

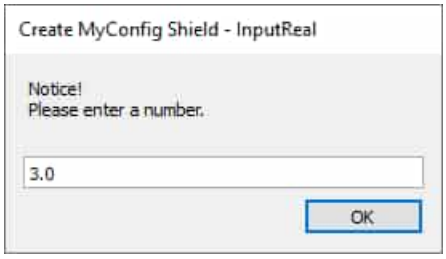


Figure 6-25 Example "Representation of a line break"

Example representation of an underlined header

```
up.underline=InputReal("<u> Notice! </u> Enter a number.",3.0)
```

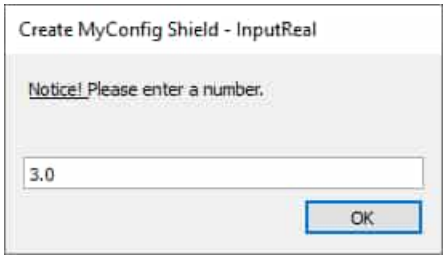


Figure 6-26 Example "Underlined header"

Example representation of a header in bold

```
up.bold=InputReal("<b> Notice! </b> Enter a number.",3.0)
```

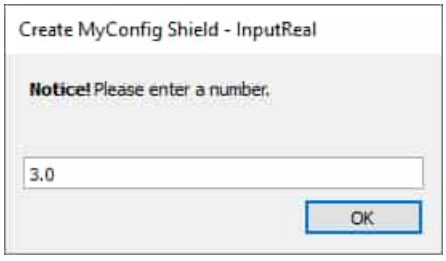


Figure 6-27 Example of a header in bold

Example representation of a header in italics

```
up.cursive=InputReal("<i> Notice! </i> Enter a number.",3.0)
```

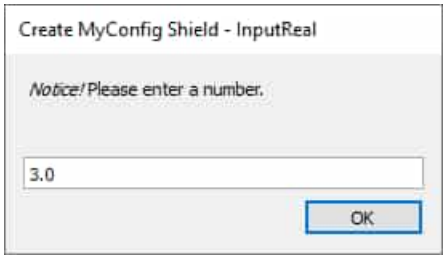


Figure 6-28 Example of a header in italics

Example representation of a bigger header

```
up.big=InputReal("<big>Notice!</big> Enter a number.",3.0)
```

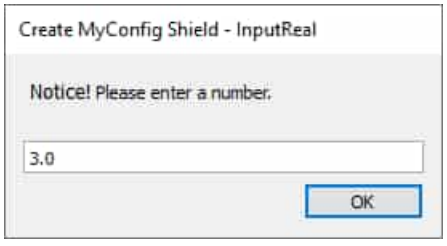


Figure 6-29 Representation of a bigger header

Example representation of a header, big and in bold

```
up.header=InputReal("<h1>Notice!</h1>Enter a number.",3.0)
```



Figure 6-30 Example of a header, big and bold - on its own line

Example of a header in another font

```
up.type=InputReal("<tt>Notice! </tt>Enter a number.",3.0)
```

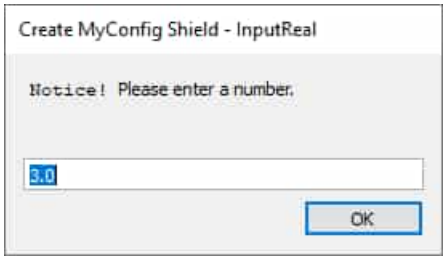


Figure 6-31 Example of a header in another font

Example of a small header

```
up.small=InputReal("<small>Notice!</small>Enter a number.",3.0)
```

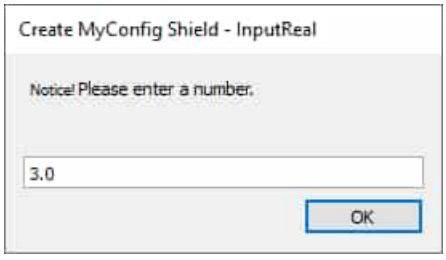


Figure 6-32 Example of a small header

Example of a header, struck through

```
up.scratch=InputReal("<s>Notice!</s>Enter a number.",3.0)
```

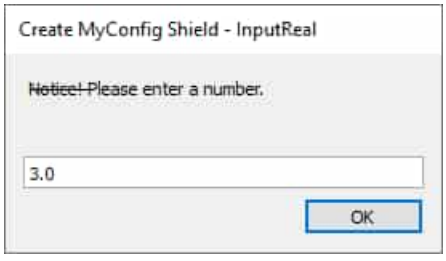


Figure 6-33 Example representation of a header, struck through

Example of an empty line between the header and the text

To display a warning, for example, you have the option of improving the representation by using an empty line.

```
up.lines=InputReal("<p>Notice!</p>Enter a number.",3.0)
```

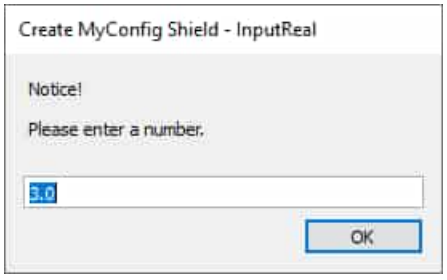


Figure 6-34 Example of an empty line between the header and the text

Example of a header in superscript

```
up.sup=InputReal("<sup>Notice!</sup>Enter a number.",3.0)
```

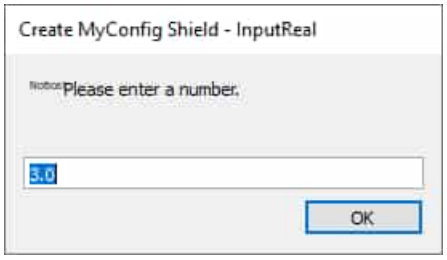


Figure 6-35 Example of a header in superscript

Example of a header in subscript

```
up.sub=InputReal("<sub>Notice!</sub>Enter a number.",3.0)
```

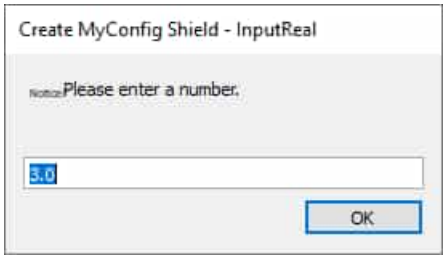


Figure 6-36 Example of a header in subscript

Examples of colored headers

Some examples of colored texts are shown below.

Header red bold

```
up.red=InputReal("<b style='color:red'>Notice!</b><br>Enter a
number.",3.0)
```

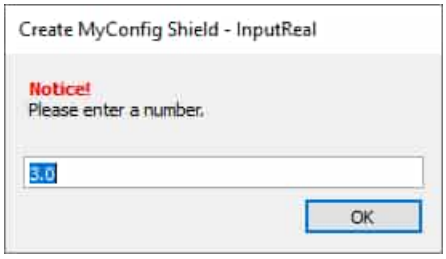


Figure 6-37 Example of a red bold header

Green header

```
up.green=InputReal("<font style='color:green'>Notice! </
font><br>Enter a number",3.0)
```

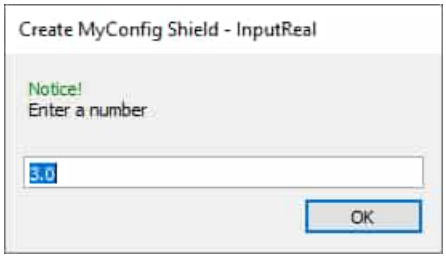


Figure 6-38 Example of a green header

Header red bold and text with yellow background

```
up.red=InputReal("<b style='color:red'>Notice!</b><br><font
style='background-color:#FFFF99'>Enter a number.<font>",3.0)
```

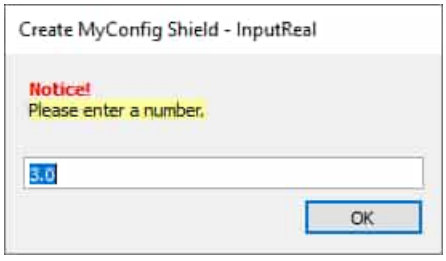


Figure 6-39 Example of a red bold header / text with yellow background

Example of background color

Header and text have different background colors. Red and green are used in the example.

```
up.background=InputReal("<b style='background-color:red'> Notice!
</b style='background-color:red'><b style='background-color:green'>
Enter a number. </b style='background-color:green'>",3.0)
```

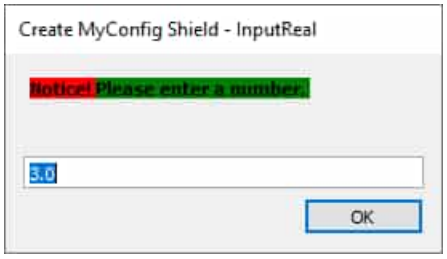


Figure 6-40 Example of the representation of text with a background color

Example of "Invisible table"

By using invisible tables, you have the option of laying out your texts.

`table border='1'` makes the table visible.

```
up.table=InputReal("<br>Select the technology!<br><table border='0'
align='center' cellpadding='6'><tr><td align='left'>turning</
td><td> 1 </td></tr><tr><td align='left'>milling</td><td> 2 </td></
tr><tr><td align='left'>grinding</td><td> 3 </td></tr></table>",1)
```

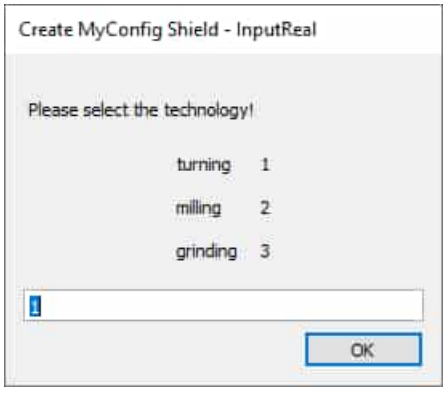


Figure 6-41 Example of "Invisible table"

Example of special characters

Use the HTML code special characters for the representation.

The special characters "less than" and "greater than" are used in the example:

```
up.special=InputReal("Notice! Enter a number that is &lt;5 and
&gt;2.",3.0)
```

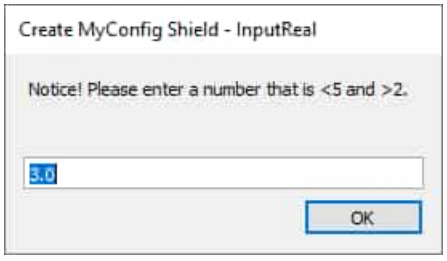


Figure 6-42 Example of special characters

More detailed information on the HTML code can be found in the appropriate references or in the Internet under HTML code.

6.21 Miscellaneous

6.21.1 Return

Return()

The `Return()` procedure can be used in all scripts to terminate the current script execution.

The operating principle of the `Return()` procedure for dialog scripts, scripts in steps and action scripts is described in Expert in the sections under:

- Dialog scripts (Page 242)
- Scripts in steps (Page 336)
- "Actions" > "Action scripts (Page 356)"

Using the procedure in a "CMC script in Diff (Page 151)" also terminates the execution.

6.21.2 ExtCall

ExtCall("<path>")

The `ExtCall()` procedure calls an external CMC script.

In principle, procedure `ExtCall()` can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

For this task, no "check run" is performed in Expert and therefore the syntax is not checked.

The file must be stored in UTF-8 coding. You can specify the path either in absolute or relative terms.

A relative path refers to the folder from which the package is executed.

An absolute path must correspond to the conventions of the runtime system.

Example

```
Extcall("./Data_X_Axis.upscr")
```

The corresponding script can be edited in the integrated "Script Editor" of Expert.

Note

Using third-party editors for CMC

A third-party editor can be used for CMC. Make sure that you use ASCII encoding or UTF-8 encoding (without BOM) when saving the file. Use UTF-8 encoding as soon as you use accented characters and special characters outside the ASCII 7-bit character set in the code characters.

6.21.3 Redo

Redo()

The `Redo()` procedure is used in dialog scripts for "OnNext" and "OnEnd" to repeat the inputs in the dialog or the complete dialog processing.

Note

"OnInit" action

OnInit is not supported in "dialog scripts", in "action scripts" and in "CMC scripts in Diff".

The `Redo()` procedure should preferably be used in the dialogs in the "OnNext" event for monitoring manual entries.

More information

More information can be found in Expert in the following section:

- "Dialog scripts" > "Executing scripts in the dialog pages" (Page 242)

6.21.4 Skip

Skip()

You use the `Skip()` procedure in "dialog scripts" to terminate a dialog.

The use of `Skip()` in a script in the step tree causes the step with the associated actions and substeps to be terminated. The execution continues at the same level in the following step.

In "Action scripts" you use `Skip()` to skip an action.

Using `Skip()` in a "CMC script in Diff" results in the script being terminated. The principle of operation is identical to `Return()`.

More information

More information can be found in Expert in the following sections:

- Dialog scripts (Page 242)
- Scripts in steps (Page 336)
- "Actions" > "Action scripts (Page 356)"

More information can be found in Diff in the following section:

- CMC scripts in Diff (Page 151)

6.21.5 DateTime

DateTime("<dt>")

The `DateTime()` function allows the determination and use of the current date and the current time.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts in steps (Page 336), "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

The `<dt>` argument informs the function about the format in which the date and time is provided. Examples are provided below.

When calling the function without an argument, i.e. with an empty string " " for `<dt>`, the date and time are formatted according to ISO 8601 "`yyyy-MM-ddThh:mm:ss`".

Also observe uppercase/lowercase within the `<dt>` argument.

Example

To generate a unique name for the logbook, proceed as follows, for example:

```
Up.dt = DateTime("yyyy-MM-dd.hh-mm-ss")
```

```
Up.$Dialog.PackageEnd.LogName = "$(Up.$Pack.DeployName)_$(Up.dt)"
```

The file extension "*.exe" or "*.usz" generates the package automatically, depending on the "runtime system" defined in the "configuration (Page 236)".

Note

You can also assign other files (archives...) with specifically generated names in this way.

Examples of units of time

```
Up.year = DateTime("yyyy") ; the year as four digit number
```

```
Up.month = DateTime("MM") ; the month as number with a leading zero
(01-12)
```

```
Up.day = DateTime("dd") ; the day as number with a leading zero (01
to 31)
Up.hour = DateTime("hh") ; the hour with a leading zero (00 to 23)
Up.minute = DateTime("mm") ; the minute with a leading zero (00 to 59)
Up.second = DateTime("ss") ; the second with a leading zero (00 to 59)
Up.milliseconds = DateTime("zzz") ; the milliseconds with leading
zeroes (000 to 999)
Up.dt = DateTime("yyyy-MM-dd hh:mm:ss.zzz") ; previous examples in
combination
```

6.21.6 DOVar

The `DOVar()` function generates DO variables during package execution in the script, irrespective of the "SINAMICS topology" dialog page.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts in steps, "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

This requires that a SINAMICS has been commissioned without any faults/errors and there is a functioning axis-drive assignment.

Use the function in packages **only** on the "Step tree" dialog page in action scripts or scripts.

Then use DO variables and their properties as usual. For example, you can work with DO variables even if there is an update in the field and the topology is "unknown".

To generate the DO variables, the user only has to know the DO name ("p199") or the permanently defined machine axis identifier (AX1, AX2...). Depending on what is known, you can use one of the two function versions described in the following.

DOVar("<doname>")

Generates a DO variable using the specified DO name `<doname>` from "p199".

Note

The DO name `<doname>` must be used with the correct uppercase/lowercase notation, as in "p199".

Example:

```
Up.myVar1 = DoVar("ALM")
```

DoVar(<axis>)

Generates a DO variable using the permanently defined machine axis identifier `<axis>`.

Example:

```
Up.myVar2 = DoVar(AX1)
```

6.21.7 Log

Log ("<label>")

The `Log()` procedure writes a user text to the logbook.

In principle, this procedure can be called in all "dialog scripts (Page 242)", scripts in steps, "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

Any message text that is only written in the logbook represents the `<label>` argument. During package execution the text is not displayed.

Example

Giving the logbook entries a "yellow" background makes the logbook easier to read.

```
Log("<span style='background-color:#FFFF99'>My notes in the CMC
logbook.</span>")
```

6.21.8 Logging

Logging (<switch>)

The procedure `Logging()` activates/deactivates logging in the logbook.

In principle, this procedure can be called in all "dialog scripts (Page 242)", scripts in steps, "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

For the `<switch>` argument, the "**On**" values are used for activating and the "**Off**" values for deactivating.

Note

Recording is automatically set back to "**On**" at the end of the script (action script/task).

Example

```
Logging (On)
```

6.21.9 MathRound

MathRound(<value>, <precision>)

You round-off values when calling the `MathRound()` function.

In principle, this function can be called in "dialog scripts (Page 242)", scripts in steps, "action scripts (Page 356)" and "CMC scripts in Diff (Page 151)".

The `<value>` argument describes the identifier whose value is to be rounded.

Examples of identifiers:

- Up variables
- Machine data
- NC data
- Drive data

Use the <precision> argument to specify the number of digits for rounding. The rounded value is assigned to the specified variable. All identifiers are permitted as variables, whose values can be derived from the corresponding result format of the function, e.g. Up variables, machine data, NC data, drive data.

Example

```
MathRound(<value>, <precision>)
         Up.pVerst = mathround(0.12345, 3) -> 0.123
         Up.pVerst = mathround(p1460[0], 2) -> 0.62
```

6.21.10 Version

Version(<area>, "<app>")

You determine the version of an installed application using function `Version()`.

In principle, this function can be called in all "dialog scripts (Page 242)", scripts in steps and "action scripts (Page 356)"; however, **not** in "CMC scripts in Diff".

The <area> argument specifies from which control area the version is determined. Possible values for <area> are NCU or PCU.

For package area PCU, all SIEMENS applications installed on the PCU, which have been entered in the PCU registry with key "`HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\AUTSW`", are determined.

For package area NCU/PPU, the versions are determined from the "`/siemens/versions.xml`" file.

The <app> argument specifies the application whose version is queried. If the version of the specified application <app> cannot be determined, then the function returns value "`null`". This means that a response is also issued when a version is not found.

For the name of the application <app>, an abbreviated, unambiguous specification is sufficient, e.g. "SINUMERIK CNC" instead of "SINUMERIK CNC-SW 31-5".

Note

All applications that are found are displayed on the "Version Display" dialog page.

6.22 From library

6.22.1 SINAMICS

6.22.1.1 SINAMICS library

Function/procedure of the "SINAMICS" library

The `SinamicsTecActivate()` and the `SinamicsTecDeactivate()` procedures are available in the "SINAMICS" library. The procedures activate or deactivate technological functions in the drive.

6.22.1.2 SinamicsTecActivate

SinamicsTecActivate("<TECNAME>", <DoVar>)

The `SinamicsTecActivate()` procedure activates technological functions of the drive.

The technological functions must be available on the SD card at the start of the CMC package. You can also install the technological functions via "Additional software (*tgz)" in the same package.

The procedure ensures that the corresponding index of the `r4956[x]` parameter is set in the servo or the `r4993[x]` parameter in the CU.

The `<TECNAME>` argument must contain the name of the technological function, in accordance with parameter `p4955[x]` (OA DO-specific Indentifier).

The `<DoVar>` argument contains the path of the DO to inform the procedure in which drive object the technological function must be activated.

The specific user parameters of the technological functions must be set independently through scripts.

In principle, this function can be called in all:

- Dialog scripts
- Action scripts
- Scripts
- CMC scripts in Diff

Examples

Example 1

Activate the "Trace buffer" function in the CU Integrated with absolute addressing.

```
SinamicsTecActivate("TRCBUF", B3_S13_PS1)
```

Example 2

Activate the "Servo Coupling" function in a servo with variable addressing.

```
SinamicsTecActivate("SERVCOUP", Up.AXSP1M_00)
```

6.22.1.3 SinamicsTecDeactivate

SinamicsTecDeactivate("<TECNAME>", <DoVar>)

The `SinamicsTecDeactivate()` procedure deactivates technological functions of the drive.

This procedures functions in the same way as the `SinamicsTecActivate()` procedure, but with deactivation of the technological functions.

6.22.2 Utilities

6.22.2.1 Utilities library

Functions of the "Utilities" library

The `IsBit()`, `SetBit()` and `ClrBit()` functions are available in the "Utilities" library. You can manipulate or query the individual bits of any variables with these functions.

6.22.2.2 IsBit

IsBit(<bit>,<value>)

The `IsBit()` function determines whether a bit is set. When the specified bit is set, the function returns the value "true" or "false".

In principle, the `IsBit()` function called is possible in all:

- Dialog scripts
- Action scripts
- Scripts
- CMC scripts in Diff

You specify which bit is to be queried via the `<bit>` argument. The bits of a value are counted from right to left starting with bit "0".

The `<value>` argument designates the identifier in whose value a bit is queried.

Identifiers are, for example:

- Up variables
- Machine data

- NC data
- Drive data
- Display data

Example

```
Up.bit_is = isBit(20,$MN_USER_DATA_HEX[0])
```

6.22.2.3 SetBit

SetBit(<bit>,<value>)

The `SetBit()` function sets a bit to the value "1".

In principle, this function can be called in all:

- Dialog scripts
- Action scripts
- Scripts
- CMC scripts in Diff

You specify which bit is to be queried via the `<bit>` argument. The bits of a value are counted from right to left starting with bit "0".

The `<value>` argument designates the identifier in whose value a bit is to be set.

Identifiers are, for example:

- Up variables
- Machine data
- NC data
- Drive data
- Display data

Example

```
$MN_USER_DATA_HEX[0] = SetBit(4,$MN_USER_DATA_HEX[0])
```

6.22.2.4 ClrBit

ClrBit(<bit>,<value>)

The `ClrBit()` function resets a bit to the value "0".

In principle, this function can be called in all:

- Dialog scripts
- Action scripts
- Scripts
- CMC scripts in Diff

You specify which bit is to be reset via the `<bit>` argument. The bits of a value are counted from right to left starting with bit "0".

The `<value>` argument designates the identifier in whose value a bit is to be reset.

Identifiers are, for example:

- Up variables
- Machine data
- NC data
- Drive data
- Display data

Example

```
$MN_USER_DATA_HEX[0] = ClrBit(10,$MN_USER_DATA_HEX[0])
```

Create MyConfig - Diff 7

7.1 Introduction

Note

Documentation structure: Operating Manual / Online Help

The contents of Section **Diff** are identical in the Operating Manual and the online help.

Diff is software for displaying and comparing SINUMERIK archives, folders, general and special files of the SINUMERIK ONE controller.

You can also change and save SINUMERIK archives with Diff.

Selected data can be exported from SINUMERIK archives or data can be imported into SINUMERIK archives.

Diff comprises the following 3 components:

- SINUMERIK archive comparison (Page 108) → Display and comparison of NC, drive or display data from SINUMERIK archives or files.
- Folder comparison (Page 154) → Display and comparison of folders, SINUMERIK archives and archives in ZIP, TGZ and TAR format.
- File comparison (Page 175) → Display and comparison of files.

The 3 software components of Diff can be called individually and any number of times in the application and are each displayed as a tab.

The Diff software can be used to support the configuration of packages with Expert, as well as standard comparison software, independently of Expert.

In all 3 components, you can edit the opened archives, folders or files.

The data determined with Diff is provided in such a way that it can be incorporated directly into a project.

The advantage of Diff that makes it so unique is the fact that it can process SINUMERIK archives, special SINUMERIK files and SINUMERIK data structures.

Diff shows NC and drive data in a structured fashion in the SINUMERIK archive comparison.

Note

SINUMERIK ONE archives

You can use ARC and DSF archives in SINUMERIK ONE.

Note

Representation of non-standard values

Diff will only display non-standard values without the option for editing.

However, it is possible to copy the machine data from another archive. To do this, copy the value from the archive, and insert the value in Diff in the "Edit mode" under tab "Selective". Then change the value.

This means that machine data in the archive can be changed.

7.2 SINUMERIK archive comparison

7.2.1 General information on the SINUMERIK archive comparison

Any number of comparison objects, e.g. SINUMERIK archives or sub files, can be imported, displayed and compared in a SINUMERIK archive comparison.

The comparison is realized at the data level.

The SINUMERIK archive comparison supports the comparison of the following objects:

- SINUMERIK ONE archives in the DSF format
- Drive archives in ASCII and ACX format
- NC/drive/display data files (INI, OPT, TEA...) in ASCII format
- Data nodes within a comparison object

The SINUMERIK archive comparison extracts the NC, drive, and display data from the SINUMERIK archives and displays this data in a structured fashion.

Unlike file comparison involving INI files, data is not compared line by line, but according to its identifiers. If the same identifiers occur a multiple number of times, these are only displayed once. The value of the last identifier that was imported is used.

The SINUMERIK archive comparison can be performed between different objects opened for comparison, but also within one object (selective).

The comparison result is prepared in a structured fashion.

You can copy data from the objects being compared to the clipboard and use it in other applications (Expert, ...).

You can remove or insert identifiers; you can also change the values of existing identifiers for SINUMERIK archives.

Diff allows an open SINUMERIK archive with modified data to be again saved as SINUMERIK archive.

Note

In the SINUMERIK archive comparison, Diff always displays all data contained in the comparison object if no settings were made in the Configure comparison to exclude data (Expert list or filter).

For SINAMICS drive data, there are parameters (r or p) that can be displayed online on the HMI. However, they are not part of an archive and therefore cannot be displayed by Diff.

7.2.2 Overview of user interface

7.2.2.1 Main menus

The main menus in Diff are identical. The submenus are dynamically adapted to the associated "SINUMERIK archive comparison", "Folder comparison" or "File comparison" component.

The following main menus and submenus are provided in the SINUMERIK archive comparison:

- File
- Edit
- View
- Comparison
- Tools
- Window
- Help

File

Table 7-1 Diff - SINUMERIK archive comparison - File main menu

File			
	SINUMERIK archive comparison		Opens a new SINUMERIK archive comparison.
	Folder comparison		Opens a new Folder comparison.
	File comparison		Opens a new File comparison.
	Open...	<Ctrl+O>	Opens an existing archive.
	Save	<Ctrl+S>	Saves an archive.

File			
	Save as...		Saves an archive with a different name or location.
	Close		Closes an archive or file.
	Archive from controller		Creates an archive from the control.
	Load into controller		Loads an archive into the control.
	Exit	<Alt+F4>	Exits the Diff program.

Note

Archive from controller / Load into controller

To create an archive from the control or load an archive to the control, you are prompted to set a password for the "Manufacturer NC access level" via the "Set user" dialog.

When the password has been entered once and the "Save password" check box activated, the password is stored in the Windows password safe.

You therefore do not have to enter a password for subsequent actions.

You can find the storage location of your password here:

Windows 10: "Control Panel" ⇒ "User Accounts" ⇒ "Credential Manager"⇒ "Windows Credentials" ⇒ "Generic Credentials"

Edit

Table 7-2 Diff - SINUMERIK archive comparison - Edit main menu

Edit			
	Cut	<Ctrl+X>	Removes the currently selected objects and copies them to the clipboard.
	Copy	<Ctrl+C>	Copies the currently selected objects to the clipboard.
	Paste	<Ctrl+V>	Pastes the objects from the clipboard to the selected position.
	Delete	<Del>	Deletes the currently selected objects.
	Select All	<Ctrl+A>	Selects all lines of a folder or data node.
	Modify	<F2>	You can changes the value of a line, provided Edit mode is activated and editing is permissible here.

Edit			
	Edit mode	<Ctrl+E>	You can switch the Edit mode on and off.
	Go to next difference	<Ctrl+N>	You can display the differences in forward or backward direction.
	Go to the previous difference	<Ctrl+B>	

View

Table 7-3 Diff - SINUMERIK archive comparison - View main menu

View			
≠	Different	<Ctrl+Shift+D>	You can activate or deactivate the display of the comparison results that are different.
=	Equal	<Ctrl+Shift+E>	You can activate or deactivate the display of the comparison results that are the same.
	Not present in all comparison objects	<Ctrl+Shift+S>	You can activate or deactivate the display of the comparison results that are not present in all comparison objects.
	Modified only		The "Modified only" icon reduces the display to show only the changed values.
	Toolbars	Standard	Like the "Search", the "Standard" toolbar can be shown or hidden. The displays are activated as default.
		Search	

Comparison

Table 7-4 Diff - SINUMERIK archive comparison - Comparison main menu

Comparison				
	Refresh	<Ctrl+D>	Refresh comparison The "Refresh comparison" button is always color-highlighted when the displayed comparison result is not up to date. Restart the update of the comparison to obtain an up to date comparison result.	
	HTML protocol		HTML protocol	Clicking the HTML protocol button displays in the standard browser (Internet Explorer) the comparison result of the node selected in the data structure. You can save the result either as HTML or XML protocol.
			HTML protocol data fields as displayed	
			XML protocol	
			XML protocol data fields as displayed	
	Edit configuration...	<Ctrl+M>	The "Edit configuration..." icon changes a comparison configuration or selects a different comparison configuration. Refer to Section "Configuration (SINUMERIK archive comparison) (Page 127)" for more information.	
	Configuration selection		Standard	Switches the comparison configuration directly without dialog.
			AFE (DO-Type 10)	
			SINAMICS_CX32 (DO-Type 4)	
			SERVO (DO-Type 11)	
			SINAMICS_I (DO-Type 3)	
			CU320 (DO-Type 1)	
			SIC (DO-Type 20)	
			BIC (DO-Type 30)	

Tools

Table 7-5 Diff - SINUMERIK archive comparison - Tools main menu

Tools			
	Options...	Opens the "Options" dialog with the tabs described below. The options can be activated or deactivated for display/execution of the Diff components. All options are **activated** as **default**. The Diff coloration and the language setting can be changed user-defined.	
		General	You can deactivate or reactivate the following settings: • Recompare objects automatically when the comparison configuration changes • Display quick start when Diff starts
		Folder comparison	You can deactivate or reactivate the following settings: Automatic selection of comparison configuration • Select the "SINUMERIK archives" configuration for the SINUMERIK archive comparison • Message for automatic configuration change Compare NCU folders with different names in the archive as having the same name. • (NCU_* == NCU_*) Restart required
		File comparison	You can deactivate or reactivate the following settings: Automatic selection of comparison configuration • Identify the file type and select the configuration • Message for automatic configuration change **File identification configuration...** Clicking this button opens the "Configuration - file identification" dialog. More detailed information can be found in Section "Automatic file identification configuration (Page 195)."
		SINUMERIK archive comparison	**Parameter**

Tools			
			You can deactivate or reactivate the following setting: • Display, copy and export with parameter short name
		Colors	The color setting for the colored line and value identification can be changed user-defined. A change of the language setting is useful, for example, for a red-green sight impediment. You can restore the standard settings as follows. • Unequal - Red • Equal - Black • Right - Blue • Left - LimeGreen • Not present in all comparison objects - DarkGrey
		Language	Language selection of the Diff interface. Available languages are: • German • English The changed language selection is activated at the next restart of the software.

Tools			
	Data comparison	Data comparison configuration...	Opens the "Configuration - NC data comparison" dialog. Information can be found in Section "Configuration - NC data comparison (Page 127)".
		Configuration data task	Opens the "Configuration - data task" dialog. Information can be found in Section "Data tasks configuration (Page 140)".
	Folder comparison	Folder comparison configuration	Opens the "Configuration - folder comparison" dialog. Information can be found in Section "Configuration - folder comparison (Page 169)".
	File comparison	File comparison configuration...	Opens the "Configuration - file comparison" dialog. Information can be found in Section "Configuration - file comparison (Page 189)".
		File identification configuration...	Opens the "Configuration - file identification" dialog. Information can be found in the "Automatic file identification configuration (Page 195)" section.
	Export configurations		The following configuration types and the associated files can be selected in the "Export configurations" dialog: • Archive/folder comparison configurations • File comparison configurations • Data comparison configurations • File identification configurations • Data tasks Only the selected configurations are exported.
	Import configurations		Configurations are imported via the path selection.

Window

Table 7-6 Diff - SINUMERIK archive comparison - Window main menu

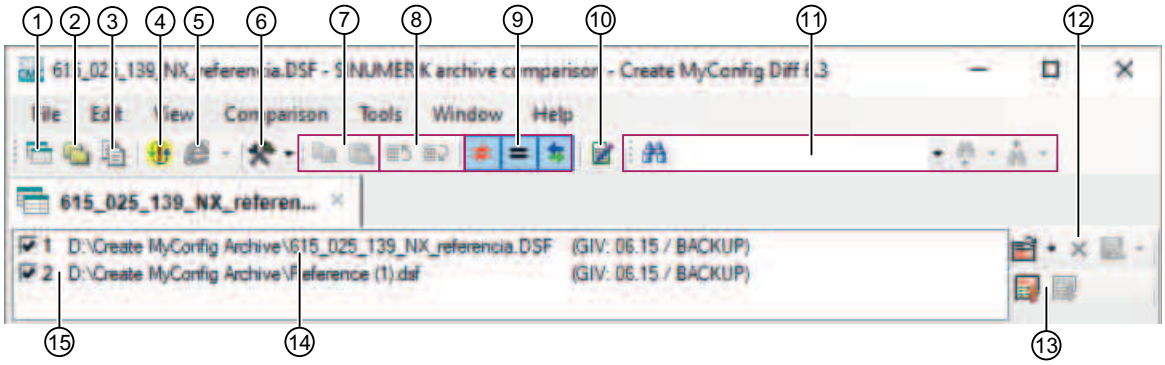
Window		
	New horizontal register group	Displays the selected component in the new horizontal register group.
	New vertical register group	Displays the selected component in the new vertical register group.
	Move to previous register group	Moves the selected component to the previous register group.
	Move to next register group	Moves the selected component to the next register group.
...	...	The display depends on the opened windows. The windows are displayed with icons for the associated component and the archive designation.

Help

Table 7-7 Diff - SINUMERIK archive comparison - Help main menu

Help			
	Content...	<F1>	Display of the online help for the **Diff** program. The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	CMC script...		Display of the Online Help "CMC script" The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	FAQs...		Display of the Online Help "FAQs" The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	Display quick start...		You open a SINUMERIK archive comparison, folder comparison, file comparison or the online help via the "Quick start" window.
	About...		Contains information on the version and copyright for Create MyConfig Diff.

7.2.2.2 SINUMERIK archive comparison - Toolbar



① New SINUMERIK archive comparison
② New folder comparison
③ New file comparison
④ Refresh comparison
⑤ Display/save HTML/XML protocol
⑥ Configure comparison
⑦ Copy / Paste
⑧ Go to previous / next difference
⑨ Different / Equal / Not available in all comparison objects
⑩ Modified only
⑪ Search functions
⑫ Select archive or file for SINUMERIK archive comparison / Close archive or file / Save as
⑬ Create archive from control / import into the control
⑭ List of opened comparison objects
⑮ Switch between opened comparisons via tab

Figure 7-1 SINUMERIK archive comparison - Toolbar

Description of buttons/fields of the toolbar

The buttons of the toolbar are described in more detail below.

① SINUMERIK archive comparison

Calls a new SINUMERIK archive comparison.

② New folder comparison

Calls a new folder comparison.

③ New file comparison

Calls a new file comparison.

④ **Refresh comparison**

The "Refresh comparison" button is always shown in color if the displayed comparison result is not up to date. This indicates that the comparison must be restarted via this button.

Note

Depending on the comparison situation, the comparison is automatically started by Diff in some cases.

⑤ **Display/save HTML/XML protocol**

This button displays the comparison result of the node selected in the data structure in the standard browser (Internet Explorer).

You will find more information in Section "SINUMERIK archive comparison - Overview window (Page 120)", position numbers ① and ②.

You can save the result either as HTML or XML protocol.

It can be output complete with expanded fields or as it is shown in the display.

⑥ **Configure comparison**

Using this button, you can change a comparison configuration or select a different comparison configuration.

More information can be found in Section "Configuration (Page 127)".

⑦ **Copy / Paste**

Buttons for copying and pasting (selective mode) selected values of one or more comparison objects.

You will find more information in Section "Copying/exporting data from the SINUMERIK archive comparison (Page 134)".

⑧ **Go to the previous or next difference.**

You can display the differences in forward or backward direction.

⑨ **Different / Equal / Not available in all comparison objects**

You can display or hide these buttons:

- Equal or unequal components of the comparison result
- The components that are not available in all comparison objects

⑩ **Modified only**

The display can be reduced to just show the modified values when this button is activated. This selection acts in addition to the display restrictions selected using button group ⑨ and those defined by the comparison configuration.

⑪ Search functions

This group contains:

- A button for opening a dialog, in which you can define the search options
- An edit box for searching for character strings in selected nodes
- Buttons for searching upward and downward


Search down


Search up

⑫ Selection of the comparison objects


Select archive or file for SINUMERIK archive comparison

The last opened archives or files are listed

Close archive or file


Button to save an archive

Save as

⑬Load archive from controller / Load into controller


Load archive from controller


Import archive into the control

Note

Set user

To create an archive from the control, or to import an archive into the control, the "Archive transfer" window opens, which allows you to connect.

You are prompted to enter your user account in dialog "Set user".

Enter your password in the "Set a password" window that then opens.

When the password has been entered once and the "Save password" checkbox activated, the password is stored in the Windows password safe.

You therefore do not have to enter a password for subsequent actions.

You can find the storage location of your password here:

Windows 10: "Control Panel" ⇒ "User Accounts" ⇒ "Credential Manager"⇒ "Windows Credentials" ⇒ "Generic Credentials"

⑭ **List of opened comparison objects**

You can open as many objects as required for comparison.

The number at the beginning of the line identifies the comparison object, e.g. in the column header.

You will find more information in Section "SINUMERIK archive comparison - Overview window (Page 120)", position number ⑧.

Note

Modified archives that have not been saved are assigned an asterisk "*" in list ⑪.

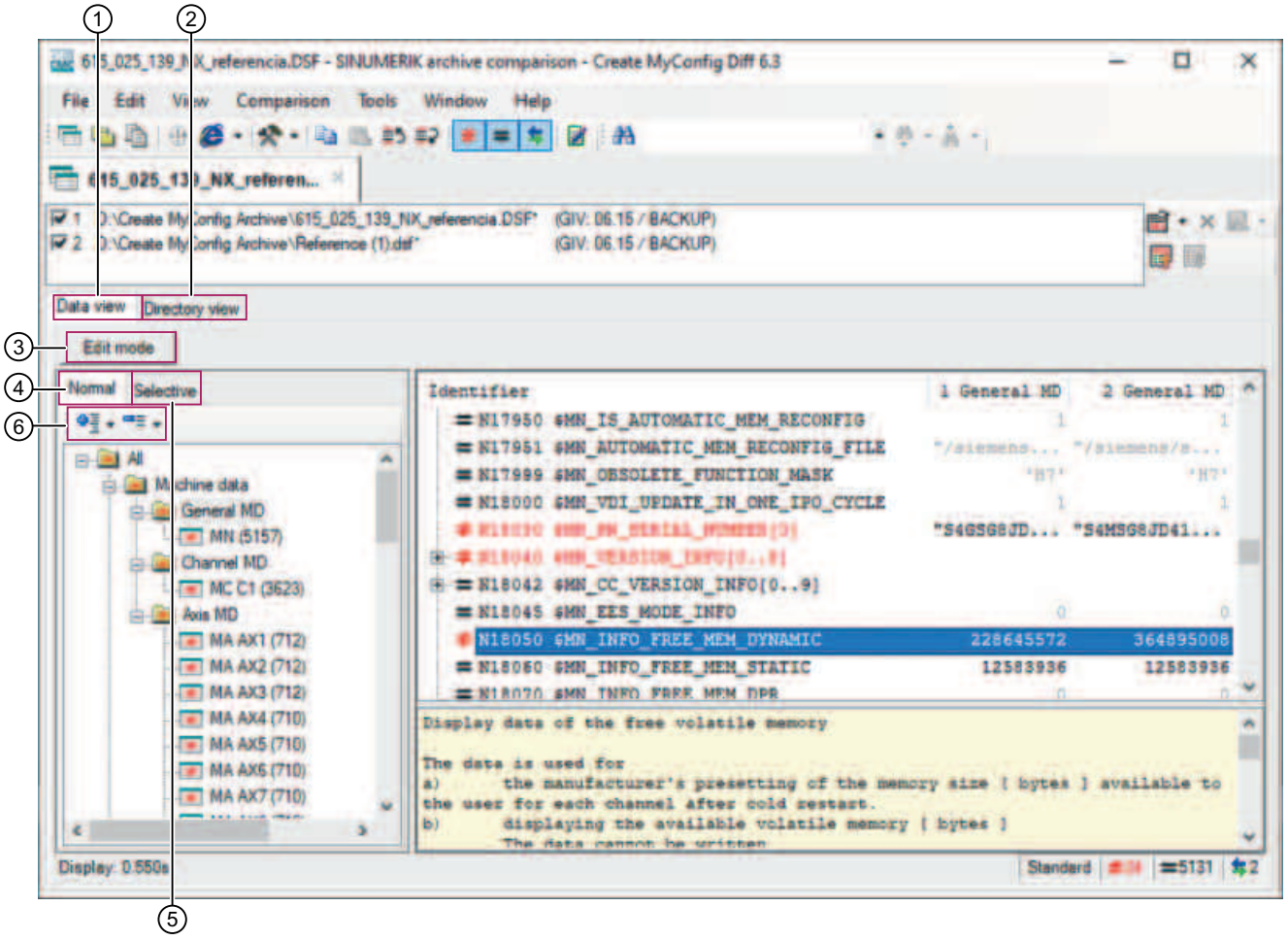
⑮ **Switch between opened comparisons**

Using the tab, you can toggle between the opened comparisons with a mouse click or by simultaneously pressing "Ctrl" and "Tab."

7.2.2.3 SINUMERIK archive comparison - Overview window

An overview of the various representations and functions in the "SINUMERIK archive comparison" window is subsequently provided.

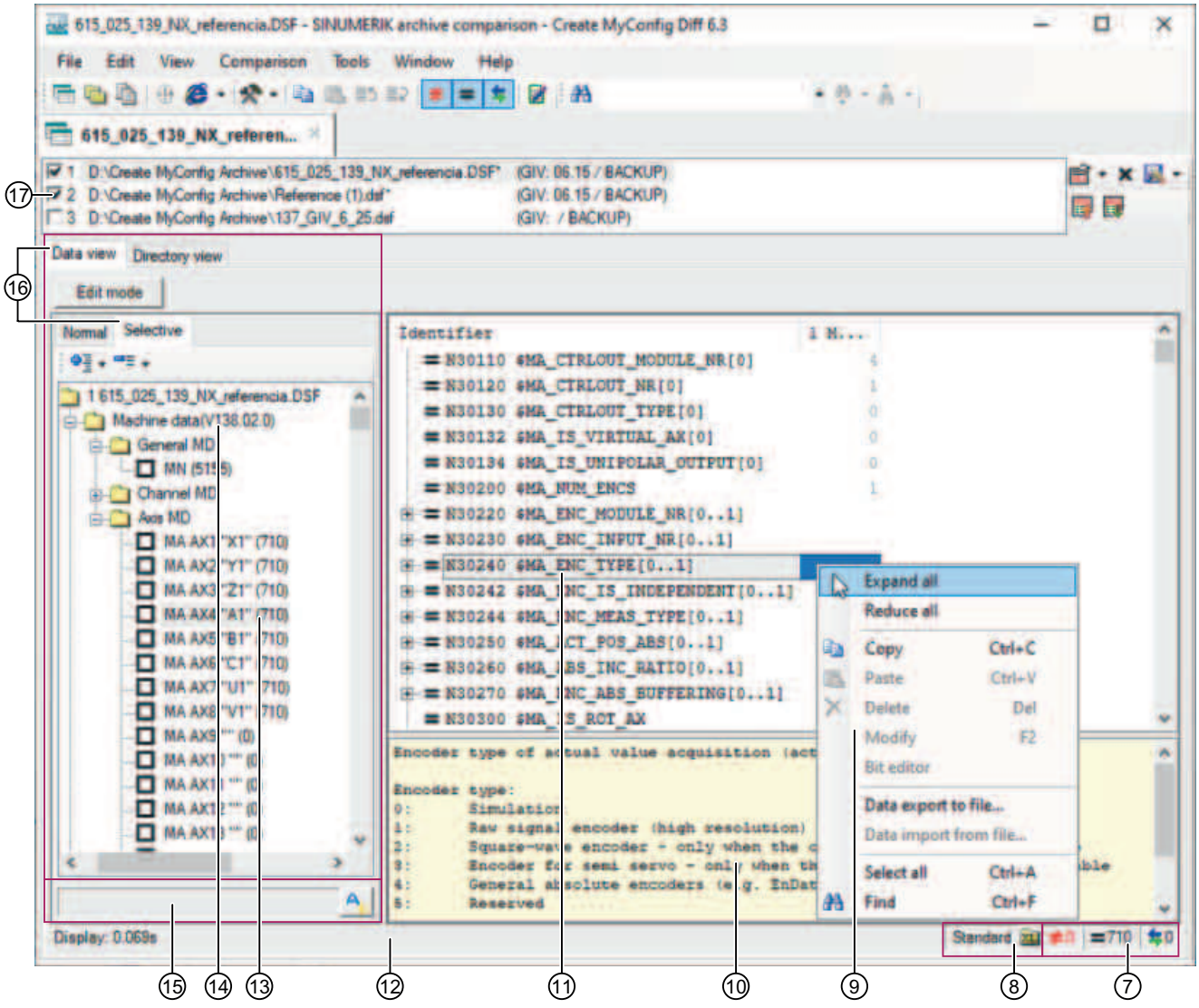
Data view tab - Normal tab



① Tab "Data view"
② Tab "Directory view"
③ Edit mode
④ Tab "Normal"
⑤ Tab "Selective"
⑥ Extend selection / Reduce selection

Figure 7-2 SINUMERIK archive comparison - tab "Data view" - tab "Normal"

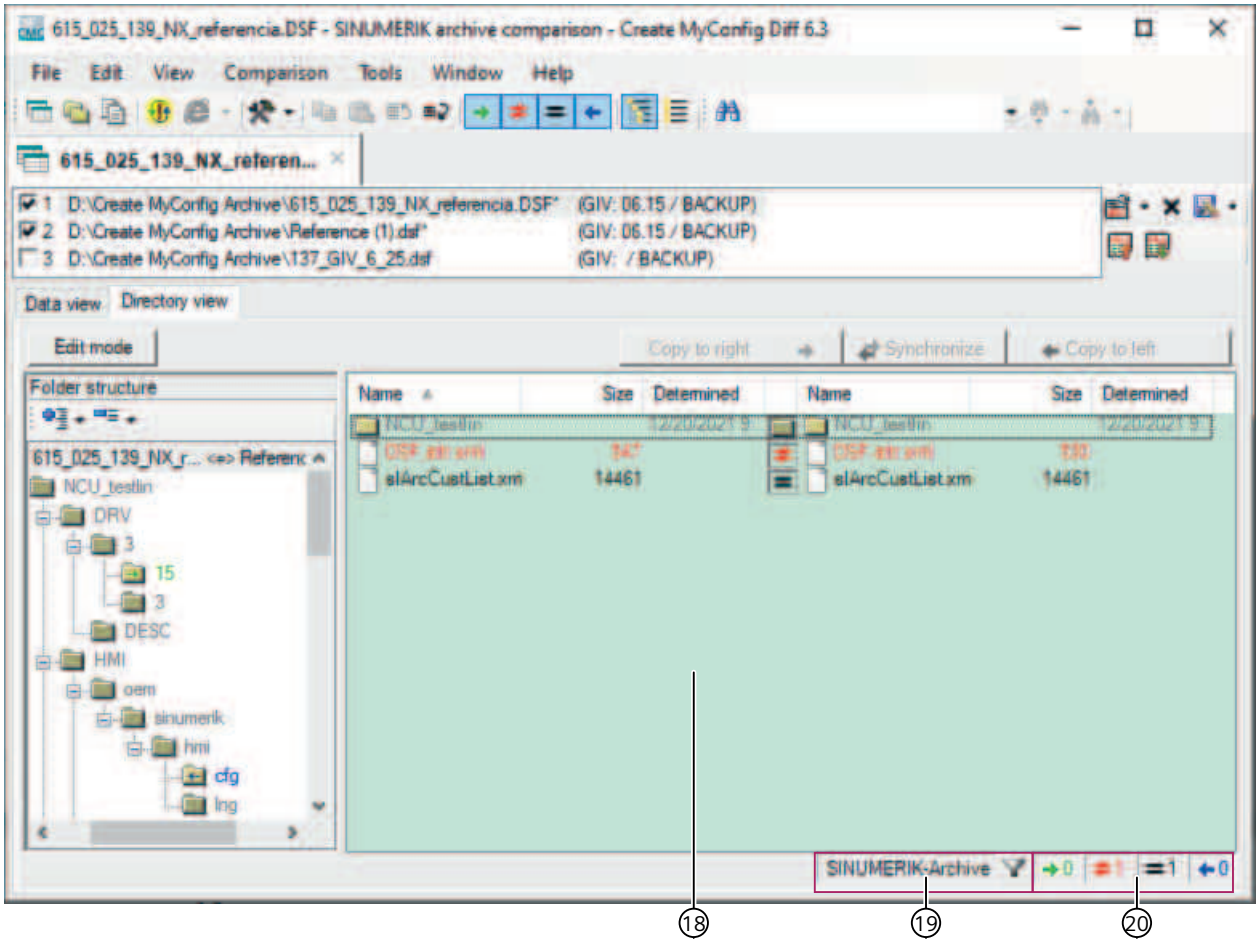
Data view tab - Selective tab



⑦ Display result: Different / Equal / Not available in all comparison objects
⑧ Configuration, expert list, filter, value comparison
⑨ Shortcut menu
⑩ Display short description for the marked parameter
⑪ Result from the "Normal" / "Selective" tab
⑫ Status bar
⑬ Short designation of the data end node
⑭ Version Display of the control area (here NC)
⑮ Axis-drive assignment

⑯ Display, tab "Data view" > tab "Selective"
⑰ Archive display area
Figure 7-3 Diff - "Data view" tab - "Selective" tab

Directory view tab



⑱ Folder comparison display area
⑲ Configuration and filters display
⑳ Comparison results display: View only left occurrences / Different / Equal / View only right occurrences
Figure 7-4 Diff - tab "Directory view"

Description of the areas/functions in the SINUMERIK archive comparison window

The individual functions of the SINUMERIK archive comparison are described in more detail below:

① Tab "Data view"

In the "Data view" tab, you can display the data in 2 different views: in the Normal or Selective mode. To do this, switch between the 2 "Normal" and "Selective" tabs.

By switching over the tab, you can set an opened archive from "Data view" to "Directory view". As a consequence, you do not have to additionally open Folder comparison.

② "Directory view" tab

The folder structure of an archive is displayed in the "Directory view".

In representation and functionality, the "Directory view" in the SINUMERIK archive comparison corresponds to the folder comparison (Page 154).

By switching over the tab, you can set an opened archive from "Directory view" to "Data view".

③ Edit mode button

The Edit mode can be activated or deactivated by clicking the button. You can only make changes, delete values and add values or identifiers in the "Data view" or "Directory view" tabs when the Edit mode is activated. You must deactivate the Edit mode to toggle between these tabs.

Note

The NC root files "*.ini" and the drive data "*.tea" cannot be edited.

④ "Normal" tab

In this area, the data nodes of the comparison objects are shown in a tree structure in the "Normal" comparison mode.

In the "Normal" comparison mode, the data structures of all comparison objects (archives, files) are placed one on top of the other and the comparison result is visualized using special node icons.

For example, the NC data of axis 1 (AX1) is always compared with the data of axis 1 or the data of drive object 2 (DO2) compared with the data of drive object 2 of another archive.

The data structure is formed from all of the data end nodes involved in the comparison of the objects.

In this mode, the end nodes of all comparison objects that are located within the selected folder are compared with one another.

Table 7-8 Meaning of the icons

- The equal character in the area node indicates that the data of the objects being compared is identical below this area node.
- The unequal character in the area node indicates that not all of the data of the objects being compared is identical below this area node.
- The equal character in the end node indicates that the data of the objects being compared is identical within the end node.
- The unequal character in the end node indicates that not all of the data of the objects being compared is identical within the end node.

The values specified in brackets after a data node specify how many identifiers of a node are displayed and how many identifiers there are in a node.

Examples:

Table 7-9 Node/identifier display

MN (1173)	This node contains 1173 identifiers, all of which are displayed.
MN (1086/1173)	This node displays 1086 of a total of 1173 identifiers. Fewer identifiers than those available are displayed because of the expert list or the filter.

⑤ "Selective" tab

In this area, the data nodes of the comparison objects are shown in a tree structure in the "Selective" comparison mode.

In the "Selective" comparison mode, the data structures of all comparison objects, such as archives and files, are placed one below the other. The structure is separately established for every comparison object.

Contrary to the "Normal" comparison mode, in the "Selective" comparison mode, all data end nodes of the same type can be compared with one another as required.

You can select the data that you wish to compare by activating the end nodes.

⑥ Extend selection / Reduce selection buttons

"Expand selection" / "Expand All": You can either display a selected folder or the complete **data view** with all subfolders and nodes in the tree structure.

"Reduce selection" / **"Reduce all"**: You can either reduce the selected folder or the complete **data view** to the main folder. As a consequence, all subfolders and nodes are then no longer visible.

⑥ Display result: Different / Equal / Not available in all comparison objects

These values provide information as to how many identifiers in the comparison results have unequal (red) or equal (black) values or which values do not exist in all comparison objects (blue/ green arrows).

⑦ Expert list and Filter configurations

If the functions "Expert list" and "Filter" are active in the comparison configuration, the relevant icons will be displayed.
Double-clicking the icons opens the expert list or filter in the comparison configuration.

Note

Using "Expert list" and "Filter," you can exclude non-relevant identifiers from the comparison, so that these are no longer displayed.

⑧ Display of the comparison configuration

The comparison configuration that is being used in the current comparison is displayed in this field.

The comparison configuration is opened with a double-click.

⑩ Display short description for the marked parameters

A brief description of the selected parameters is displayed in the text area with yellow background.

⑪Result from the "Normal" / "Selective" tab

The data of the nodes selected in areas ④ and ⑤ is prepared in a tabular form in this area of the user interface.

The identifiers are listed in the column to the far left, and in all of the additional columns, the associated values of the individual comparison objects (archives, nodes).

The comparison results are color-coded to make it easier to interpret the results:

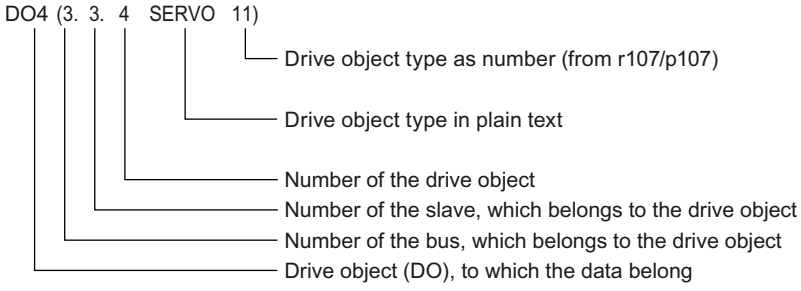
- Identifiers shown in **black** indicate that all values of the comparison object for this particular identifier are the same.
- Identifiers shown in **red** indicate that not all values of the comparison object for this particular identifier are the same.
- Values shown in **blue** are values that the user changed in the "Selective" mode.

The number as prefix in the column header of the values defines the comparison object (archive, file) to which these values belong.

⑬ Short designation of data end node

The data end nodes have a code that defines the data area from which the data originated.

For SINAMICS data, the following information is contained in the node designation.



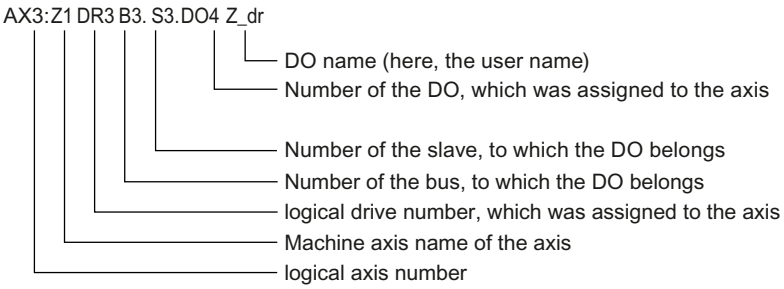
⑭ Version Display

After the node names "Machine data" and "SINAMICS", the NC and SINAMICS version are displayed from where the data originates.

⑮ Axis-drive assignment

Displaying the axis-drive assignment of a selected drive object or axis.

Example



This information is **only** completely displayed for archives that contain both NC data and drive data.

7.2.3 Configuration

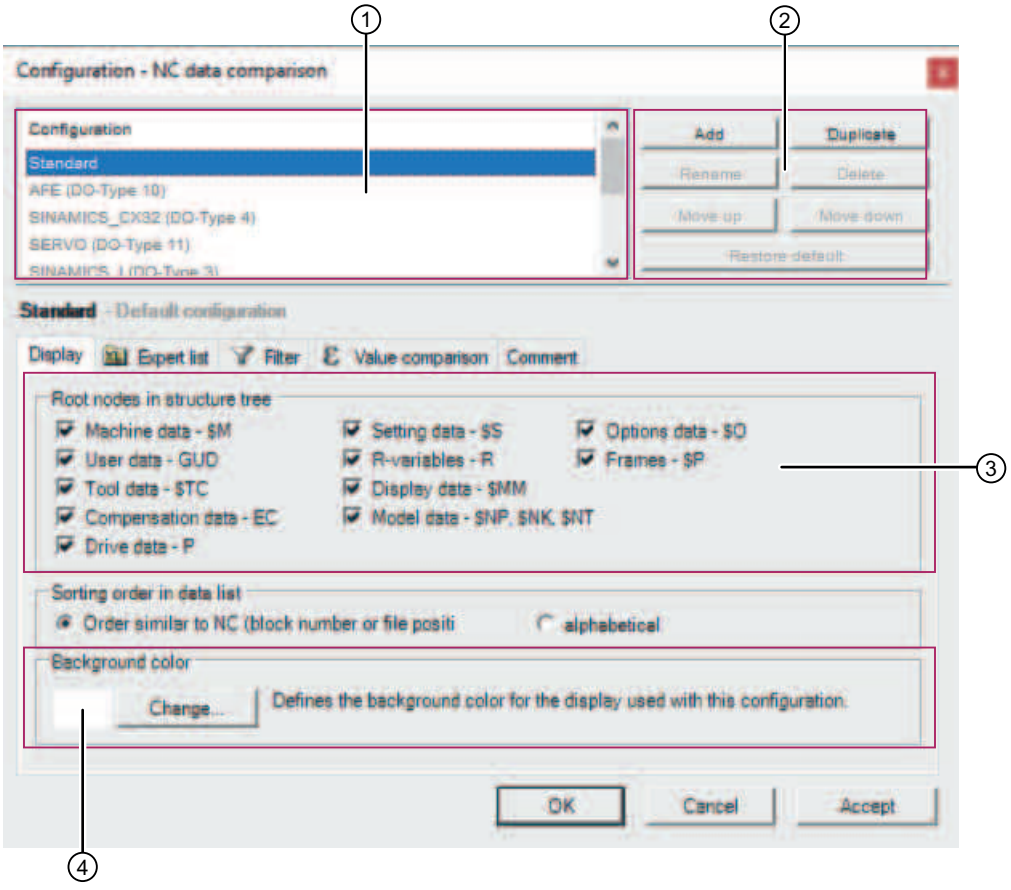
SINUMERIK archive comparison

You call the dialog to configure the SINUMERIK archive comparison using the "Configure comparison" button . You manage the various comparison configurations in dialog "Configuration - NC data comparison". You can create, edit and delete comparison configurations in this dialog.

All criteria to be considered for a SINUMERIK archive comparison are defined in a comparison configuration.

In order to provide the configurations to other users, you can export and import configurations.

More information is provided in Section "Exporting and importing configurations (Page 198)".



① List of the available comparison configurations
② Buttons for managing the configurations
③ Main nodes in the structure tree
④ Background color

Figure 7-5 Configuration - NC data comparison - tab "Display"

① List of the available comparison configurations

The existing comparison configurations are listed in this area of the dialog "Configuration - NC data comparison". You can select the required comparison configuration by clicking on it.

Configurations shown in gray are user-specified configurations; you can change these; however, you cannot delete them and you can restore them at any time

② Buttons for managing the configurations

You can manage the configurations using these buttons.

A new configuration is created via "Add", which you can adapt.

You make a copy of the selected comparison configuration via "Duplicate".

With "Restore default", the changes to the specified comparison configuration are undone.

Note

The "Restore default" button is only active if a user-specified configuration has been selected, the contents of which are no longer the same as the default setting.

This means, that in addition to restoring the default setting, this button can also be used to check whether the contents are identical with those of the default setting.

③ **Root nodes in structure tree**

You can save the properties that belong to the selected configuration in this area of the dialog; these properties include display criteria, filter criteria and a comment.

④ **Background color**

So that the user can immediately recognize how to select certain comparison configurations, you can define a separate background color for each comparison configuration.

Display tab

Under the "Display" tab, you define which main nodes should be displayed in the data structure tree, assuming that data is available in the comparison object (archive or file) for this main node.

Further, you can define whether the data should be sorted in alphabetic order or similar to that used in an NC in the data list.

Expert list tab

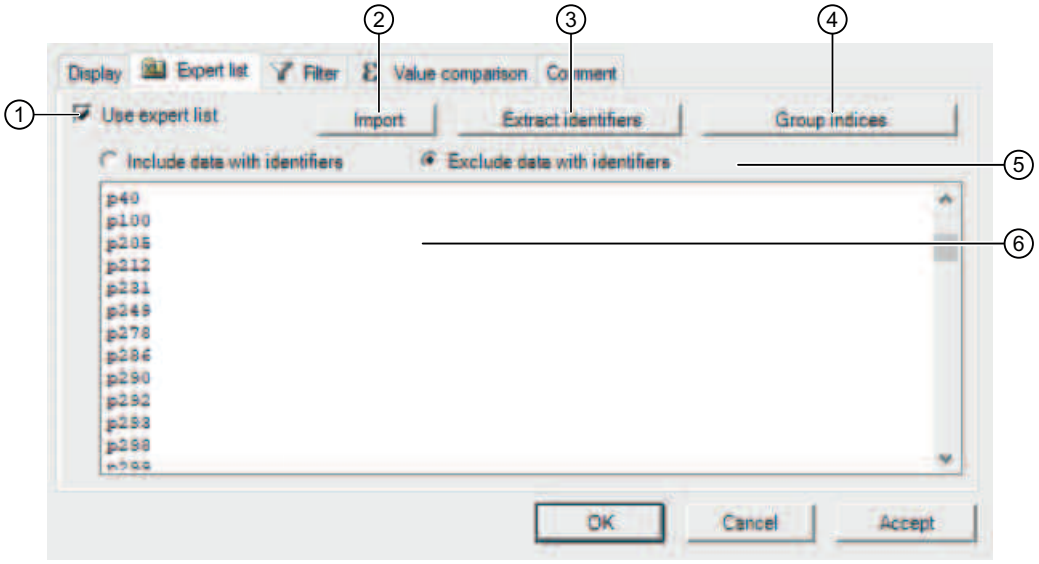
Using the Expert list, you can specify a number of identifiers that should either be included or excluded for the comparison.

Always enter the identifiers completely in the "Expert list". The only exception is the field data in the identifier.

Example:

A `$MA_CTRLOUT_NR[0]` entry in an inclusion list precisely includes this identifier.

A `$MA_CTRLOUT_NR` entry in an inclusion list includes all identifiers that belong to this field, i.e. `$MA_CTRLOUT_NR[0]` and `$MA_CTRLOUT_NR[1]`.



① Using the expert list
② "Import..." button
③ "Extract identifiers" button
④ "Group indices" button
⑤ Include/exclude data with identifiers
⑥ Field listing the identifiers

Figure 7-6 Expert list

① Use expert list

You can activate use of the expert list by selecting the option box "Use expert list".

Note

If an expert list is active, then all of the identifiers that have been filtered-out are not compared and are also no longer displayed. The icon is displayed in the status bar.

② Import...

You import the identifiers from a file using this button.

③ Extract identifiers

If you import identifiers via the clipboard or a file, then values or more information that are not identifiers can be included.

Using this button, you can extract the identifiers from the imported characters.

The "Extract identifiers" function removes the identifiers that occur more than once in the list.

Comments are not removed; this involves all the characters after a semicolon.

④ Group indices

Using this button, you can combine indexed machine data to form global machine data:

$MN_AXCONF_MACHAX_NAME_TAB[3] → $MN_AXCONF_MACHAX_NAME_TAB

The indices are removed and only one date without indices remains in the list.

The first comment on the data is retained at this data item.

⑤ Include / exclude data with identifiers

Here, you specify whether the listed identifiers and their associated data are to be included or excluded for the comparison.

⑥ Field listing the identifiers

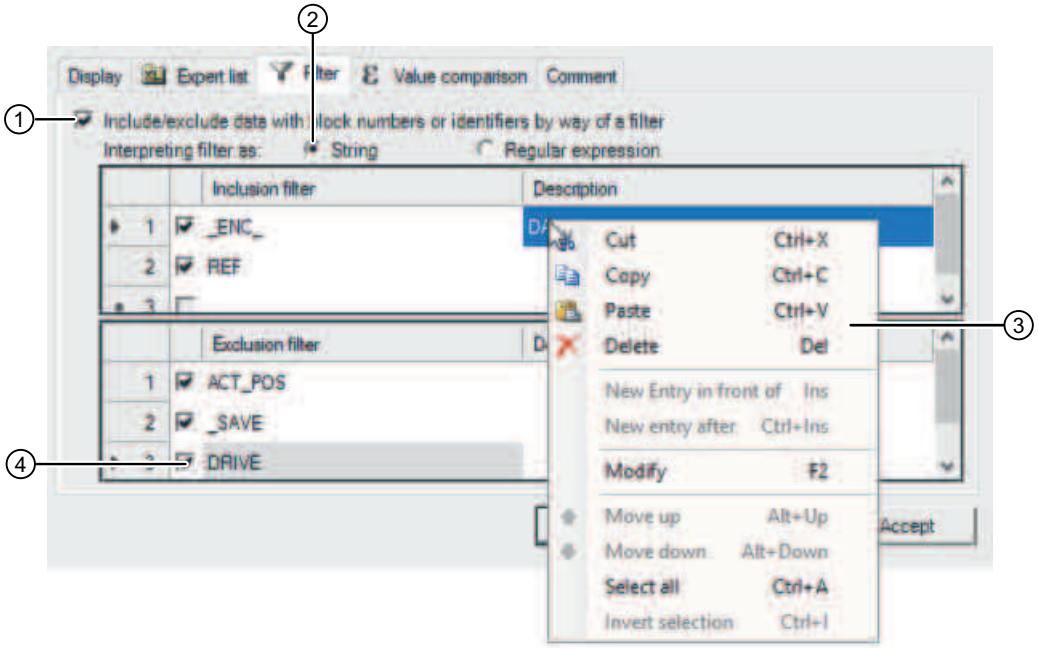
The selected identifiers are displayed in this field. There may only be one identifier in each line.

The identifiers can be imported via the clipboard or with the import function.

Filter tab

Using filters, you can include or exclude identifiers for the comparison based on character matches.

The filters are applied to the identifiers and their block number. The function is not case-sensitive.



① Activate filter
② Define filter as character string or regular expression
③ Using the shortcut menu, you can create, edit and delete entries under the "Filter" tab.
④ Every entry in the "Filter" tab can be individually activated or deactivated.

Figure 7-7 NC data comparison - filter

① Activate filter

By selecting "Include/exclude data with block numbers or identifiers by way of a filter", the functionality of the inclusion and exclusion filter is activated.

Note

If a filter is active, then the filtered identifiers are neither displayed nor compared.

The Filter icon is displayed in the status bar.

② Define filter, as character string or regular expression

Interpret filter as:

You define whether the filters should be interpreted as **character string** or **regular expression**. Information on this can be found in Section "Use of regular expressions in Diff (Page 206)".

Generally, the filters do not describe complete identifiers, but distinct character strings of the identifiers which are sufficient to clearly identify the required quantity of identifiers.

③ Shortcut menu

Using the shortcut menu, you can create, edit and delete entries under the "Filter" tab.

④ Activate or deactivate an exclusion filter

Every filter entry can be individually activated or deactivated.

Example of a filter:

SAFE	This filter describes all identifiers that contain the specified character string. Generally, these are all safety data, e.g.: `$MA_SAFE_IS_ROT_AX`, `$MA_SAFE_SINGLE_ENC`

Value comparison tab

In the "Value comparison" tab, the value-based comparison of floating-point numbers is displayed at significant digits.

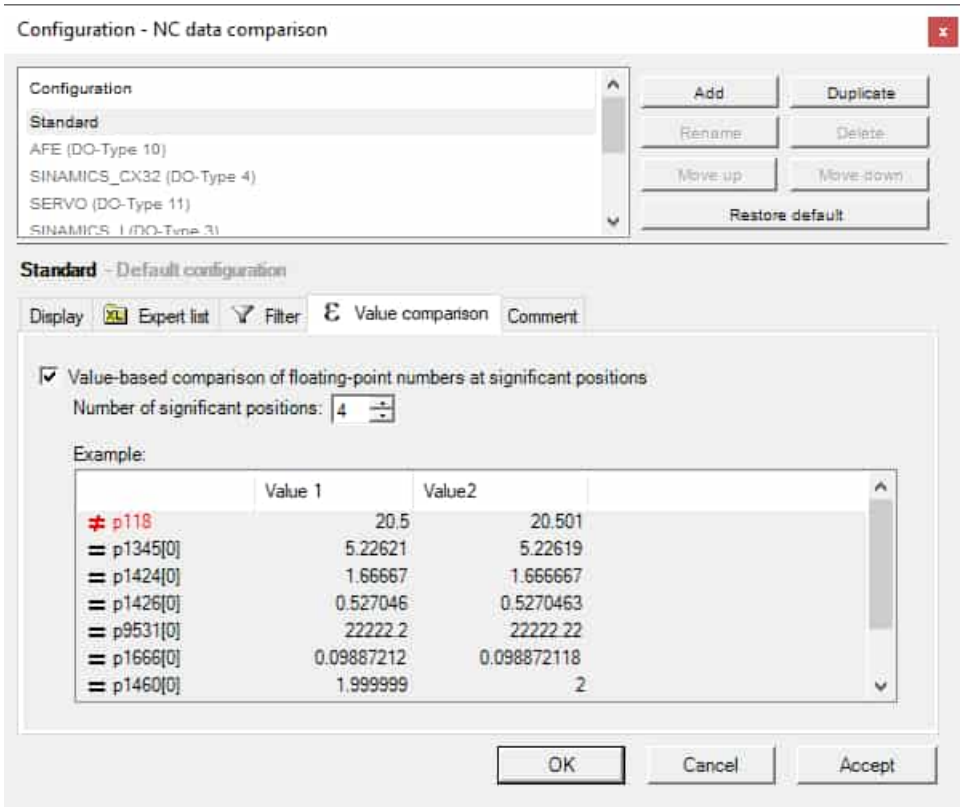


Figure 7-8 Value comparison tab

Floating point numbers are saved separately and internally in the computer in mantissa and exponent.

The exponent is adapted in such a way that the mantissa is normalized, i.e. in the range of "1 <= m < 10".

The maximum possible number of significant digits is therefore "7" for floating values and 15 for double values.

Other inaccuracies can occur as a result of parameter calculations, which can be consciously ignored during the comparison.

When the comparison algorithm is activated, the difference of the values is normalized via the absolute value of the values. If this is then less than the specified significant digit (the mantissa), the values are considered equal.

Comment tab

Any comments for the selected comparison configuration can be saved in the "Comment" tab.

7.2.4 Copying/exporting data from the SINUMERIK archive comparison

In the SINUMERIK archive comparison, data identifiers and the associated values can be copied to the clipboard or exported to a file.

For instance, you can use copied or exported data in scripts of Expert, in Excel for documentation, in another software application or also in a SINUMERIK archive comparison in Diff.

In the SINUMERIK archive comparison, depending on the data marking, there are 2 different copy and/or export formats:

- Value assignment
- Value table

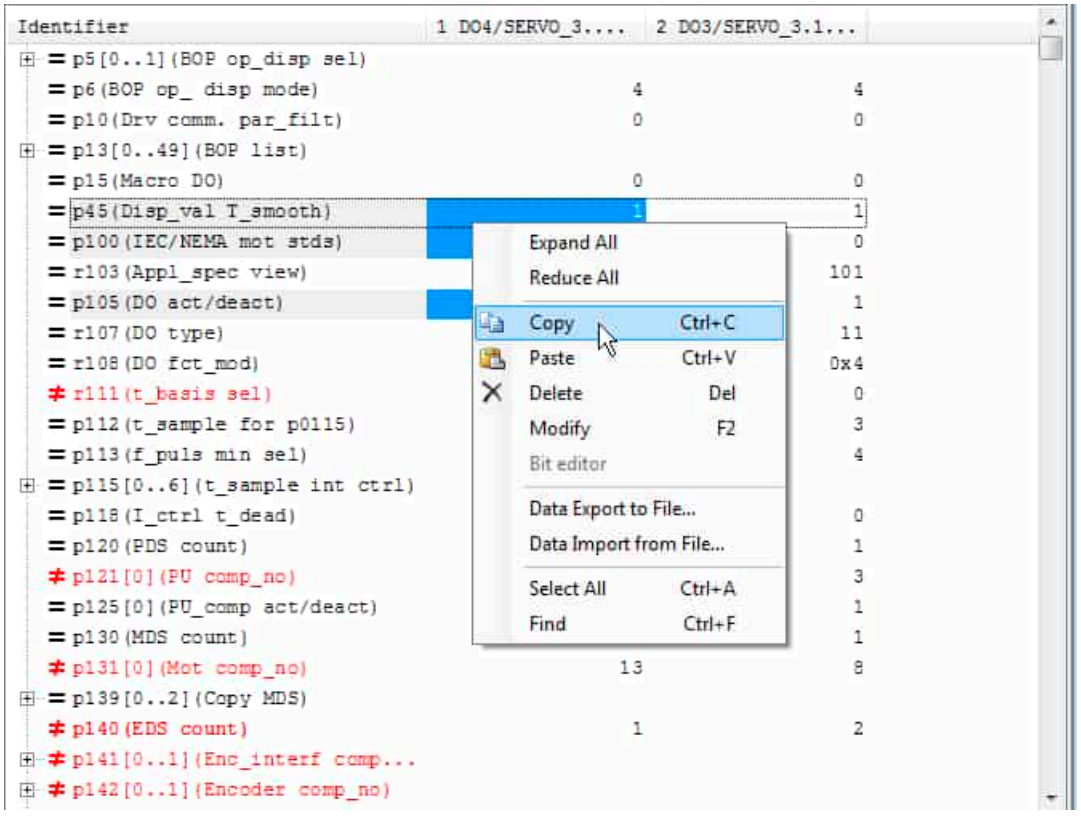
Copy/export format "Value assignment"

For the copy/export format "Value assignment", data is extracted in the form "identifier = value". This means that data is extracted with unique, fully qualified identifiers, a value for each identifier as well as information about the source data area - and is saved in a form that you can directly use in scripts of Expert.

This type of copying and exporting is the most frequently applied type.

To copy or export data, select the values in the value column as shown below.

Example 1: Copying SINAMICS data



Using the "Copy" or "Data export to file..." shortcut menu entries, you can transfer the selected values and their identifiers to the clipboard or a file.

For the selection shown, the following content is obtained for the clipboard or file:

```
[B3_S3_PS3] ;V2.40.43.00
p45=1
p100=0
p105=1
```

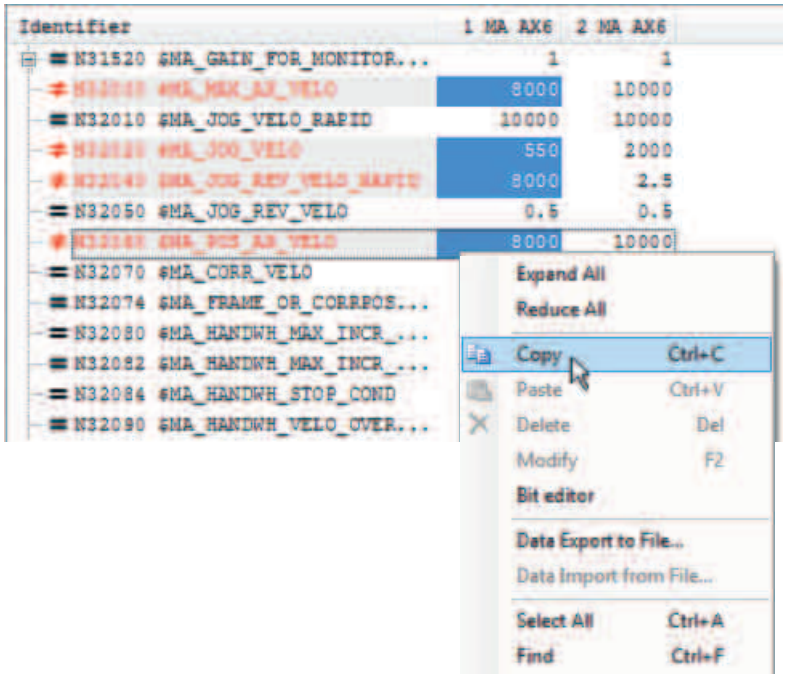
A section with the information about the data source is provided in the first line.

This information is saved in an Expert-conform format so that it can be used in a script as "Data target" for the following values.

The SINAMICS version – from which the following data comes – is saved after the section, separated by a semicolon.

The value assignments in the form "identifier=value" are in lines 2 to n.

Example 2: Copying NC axis machine data



For the selection shown, the following content is obtained for the clipboard or file:

```
CHANDATA(1)
N32000 $MA_MAX_AX_VELO[AX6]=8000
N32010 $MA_JOG_VELO_RAPID[AX6]=8000
N32020 $MA_JOG_VELO[AX6]=550
N32060 $MA_POS_AX_VELO[AX6]=8000
```

The identifiers extracted by Diff are fully qualified, i.e. they again include the definition of the axis to which the data belongs.

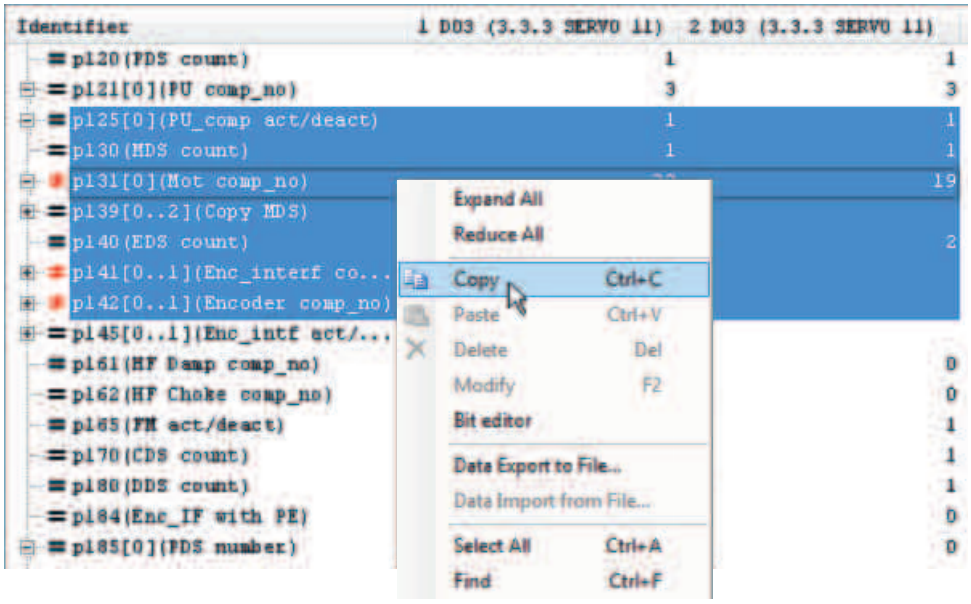
Note

CHANDATA(1) is also automatically added for channel-independent data as this guarantees that a previous channel assignment greater than "1" is again reset.
Channel-independent data must not be located below a CHANDATA instruction greater than "1".

Copy/export format "Value table"

For the copy/export format "Value table", data is extracted in tabular form. This means that all selected identifiers are saved in the first column, and in the additional columns, the values of the particular comparison object (archive, file or node).

To copy or export data in this way, select the identifiers in the Identifier column, as shown in the following example.



For the selection shown, the following content is obtained for the clipboard (to insert in Excel) or file:

Identifier	1 DO3 (3.3.3 SERVO 11)	2 DO3 (3.3.3 SERVO 11)
p125	1	1
p130	1	1
p131 [0]	22	19
p139 [0]	0	0
p139 [1]	0	0
p139 [2]	0	0
p140	2	2
p141 [0]	20	17
p141 [1]	0	0
p142 [0]	21	18
p142 [1]	0	0

7.2.5 Pasting/importing data in the SINUMERIK archive comparison

Introduction

In the SINUMERIK archive comparison, data can be pasted from the clipboard or imported from a file.

This function is available **only** in the "Selective" comparison mode and only for SINUMERIK archives (not INI/TEA files...).

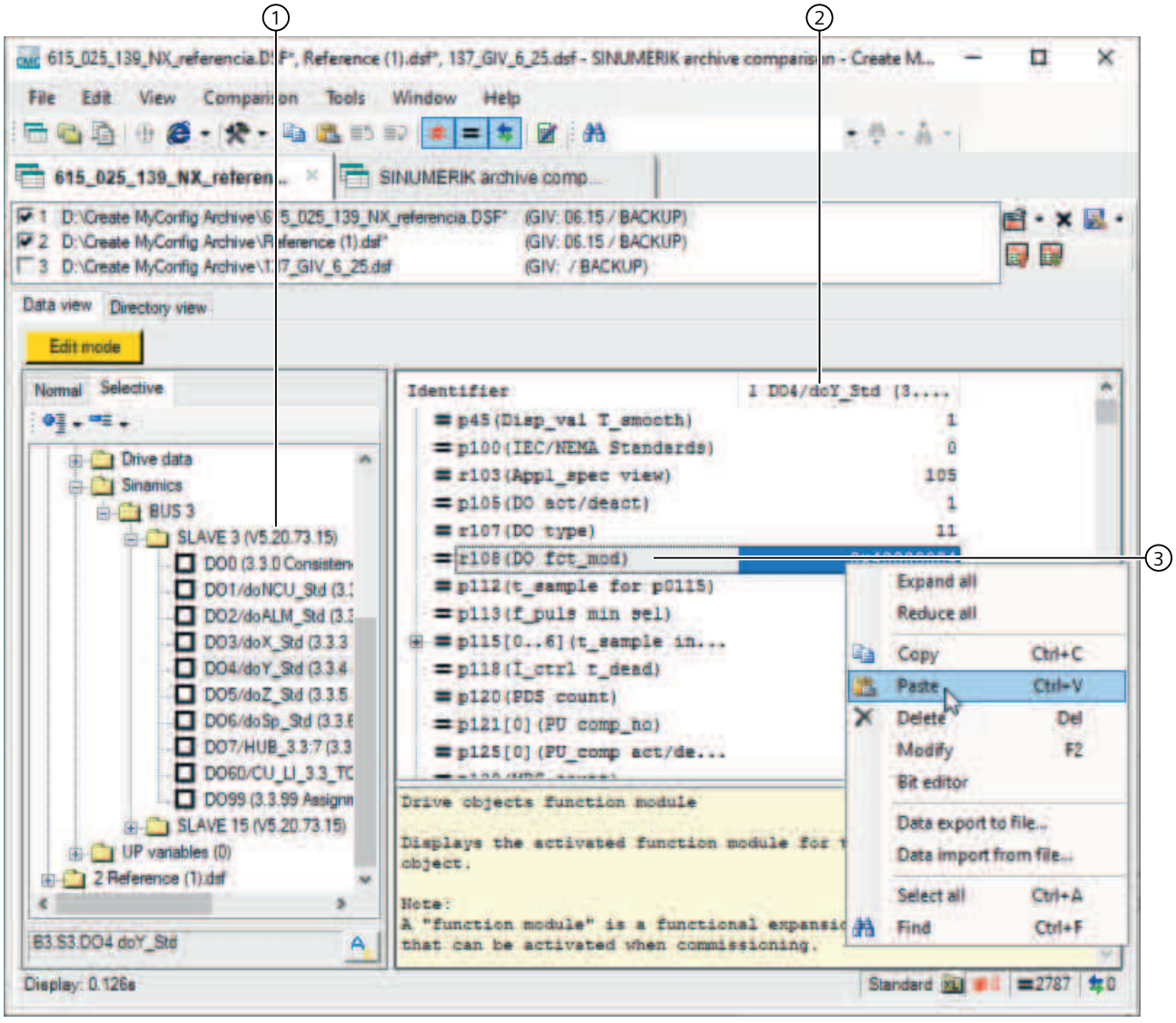
First activate the "Edit mode" button in order to add data.

Use only data in the clipboard or in a file in the form "identifier=value" (copy/export format, "Value assignment").

Note

The copy/export format "Value table" from Diff is **not** supported for paste/import.

Using the shortcut menu, it is possible to paste/import data to a value column ②.



① Selected data nodes
② Value column in which data is inserted or imported
③ Selected value

Figure 7-9 Pasting/importing data in the SINUMERIK archive comparison

When pasting/importing data, Diff searches the identifiers contained in the clipboard/file in the data nodes ① selected in the value column ② and pastes the "new" values for the identifiers found.

Data from scripts from Expert is supported for pasting/importing. Comment lines are ignored. Replacement operators are ignored when used at appropriate places, e.g. as axis index.

Note

If the number of values ③ selected in value column ② is equal to the value contained in the clipboard/file, then the selected values are overwritten and the identifiers are no longer taken into consideration.

When pasting into data nodes ① the values are always pasted, taking into consideration the identifiers.

"Preview pasting" dialog

To show what the result of pasting or importing values will be before values are actually pasted or imported, Diff displays the "Preview pasting" dialog.

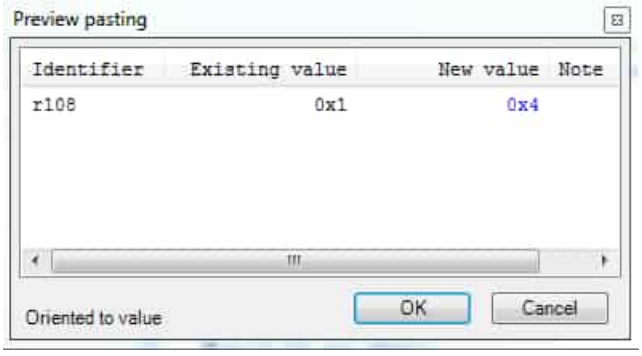


Figure 7-10 Preview pasting

The following information is shown in the dialog:

- This dialog box lists all of the identifiers with the existing and new values that are to be pasted. If, as a result of the new value, the value changes with respect to the existing value, then the new value is shown in blue.
- If no identifiers (i.e. no existing values) could be found for the values to be pasted, or if there are format problems, this is also shown in the dialog.

After pasting or importing values, changed values are identified in Diff in blue and the path of the comparison object (archive) is identified in the list of opened comparison objects using an asterisk "*". The "Save" button is still active for the comparison object.

7.2.6 Data tasks in the SINUMERIK archive comparison

7.2.6.1 Overview

In the SINUMERIK archive comparison, using data tasks, data can be converted, imported and exported.
A configuration and an Excel template belong to a data task.

Supplementary conditions for data tasks

- Data tasks are only available in the "Selective" comparison mode.
- Data tasks can only be applied to data end nodes.

Application examples for data tasks

- Importing encoder data.
  Encoder data saved in an Excel file (template) can be imported into an archive.
- Exporting optimization data.
  For the identifiers of optimization data saved in an Excel template, values can be added and saved as Excel file.

Principle of operation of import tasks

When executing import tasks, values are imported into the clipboard from the "Import area" of an Excel template.
In Diff, these values can be inserted at a data end node. When inserting values, the values of existing identifiers can be overwritten and new identifiers can be inserted.

Principle of operation of export tasks

When executing export tasks, data identifiers are determined in an "Export area" of an Excel template. These identifiers are searched for in a selected data end node in the data structure of the SINUMERIK archive comparison and their values are transferred into an Excel template.

The "Save As" dialog allows the completed Excel template to be saved.
The "Save As" dialog is displayed, which allows the completed Excel template to be saved.

Principle of operation of conversion tasks

When executing conversion tasks, data identifiers are determined from an "Export area" of an Excel template. These identifiers are searched for in a selected data end node in the data structure of the SINUMERIK archive comparison and their values are transferred to an Excel template.
The conversion rule saved in the same Excel template is processed by Excel without any visualization. In so doing, values are generated for the saved import identifiers and these are then imported to Diff from the "Import area". The imported values are saved in the clipboard, and can be inserted into Diff at a data end node.
When inserting values, the values of existing identifiers can be overwritten and new identifiers can be inserted.
The Excel file with the converted data can be saved.

7.2.6.2 Configuring data jobs

The dialog to configure data tasks is called from the "Tools" > "Data comparison" > "Configuration - data task" main menu.

The dialog to configure data tasks is equivalent to the dialog for the comparison configuration and should be used in the same way.

You can also export or import configurations, for instance, to make them available for other users.

More information on this topic is provided in Section "Exporting and importing configurations (Page 198)".

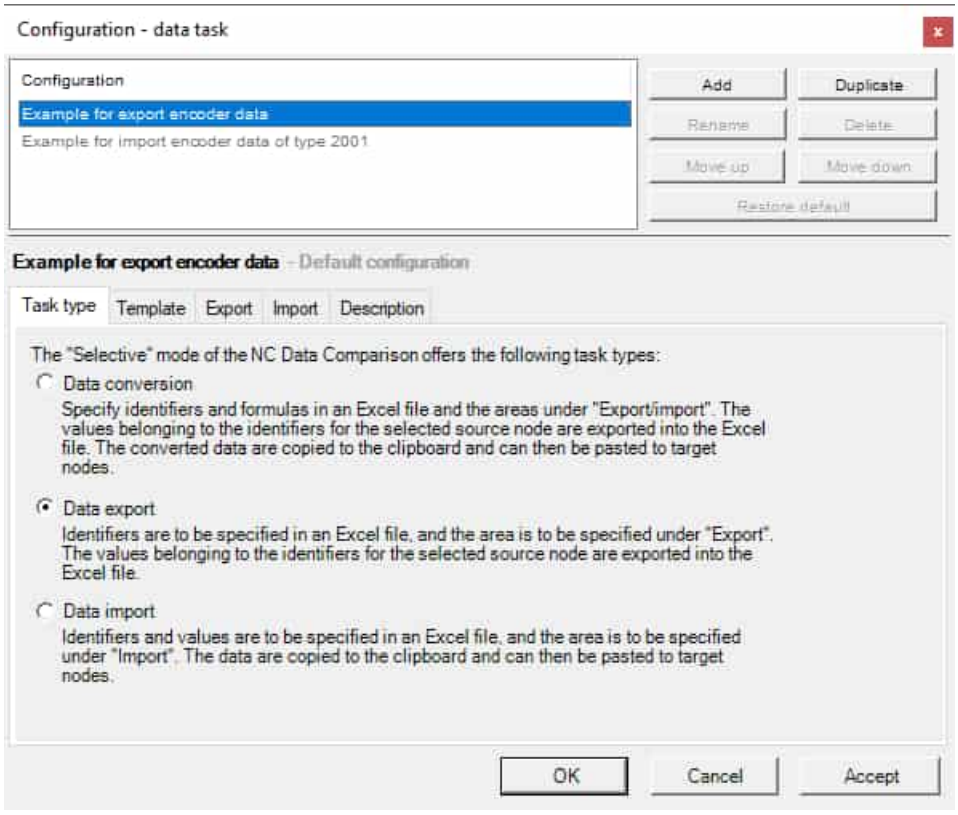


Figure 7-11 Configuration - data task

Task type tab

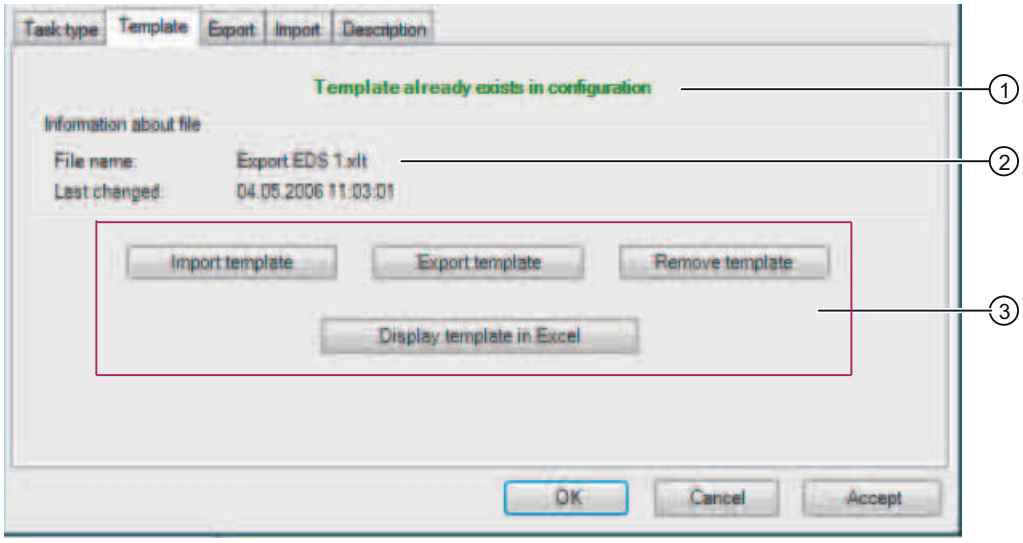
Select the task types in the "Task type" tab:

- Data conversion
- Data export
- Data import

Template tab

You can store an Excel template in the "Template" tab for the selected data task. This is required for the data conversion, but also for the data import and the data export. Diff Saves this template as a fixed component of the configuration.

If a template is not saved in this tab, then when executing the data task, Diffrequests a template to be specified via an "Open file" dialog. It makes sense not to save a template if Diff should not operate with a fixed template, but instead with an external template that is assigned during the package execution.



① Displays as to whether an Excel template exists in the configuration.

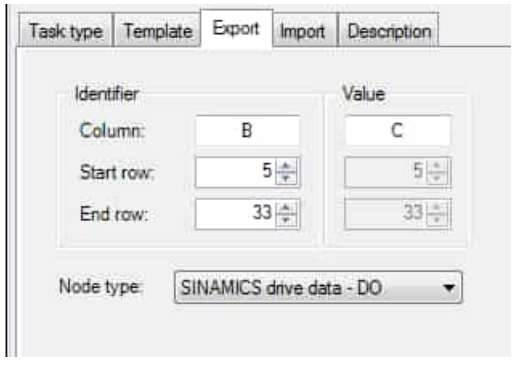
② Details of the name and date of the file used as Excel template.

③ These buttons allow the following:

- Import a file as template in the configuration
- Export the configuration contained in the template as file
- Remove the template saved in the configuration
- Display the template contained in the configuration using Excel.

Export tab

The export area of the identifiers and values are specified in the Excel template in the "Export" tab.

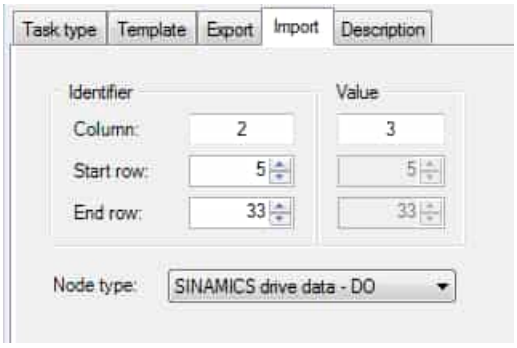


Enter a "Node type" in this tab in the data structure tree. You can only start an export or conversion task for this node type.

The "Node type" should be selected that belongs to the data to be exported.

Import tab

The import area of the identifiers and values are specified in the Excel template in the "Import" tab.



Only for an import task, enter a "Node type" in this tab in the data structure tree. You can only start an import task for this "node type".

The "Node type" should be selected that belongs to the data to be imported.

Note

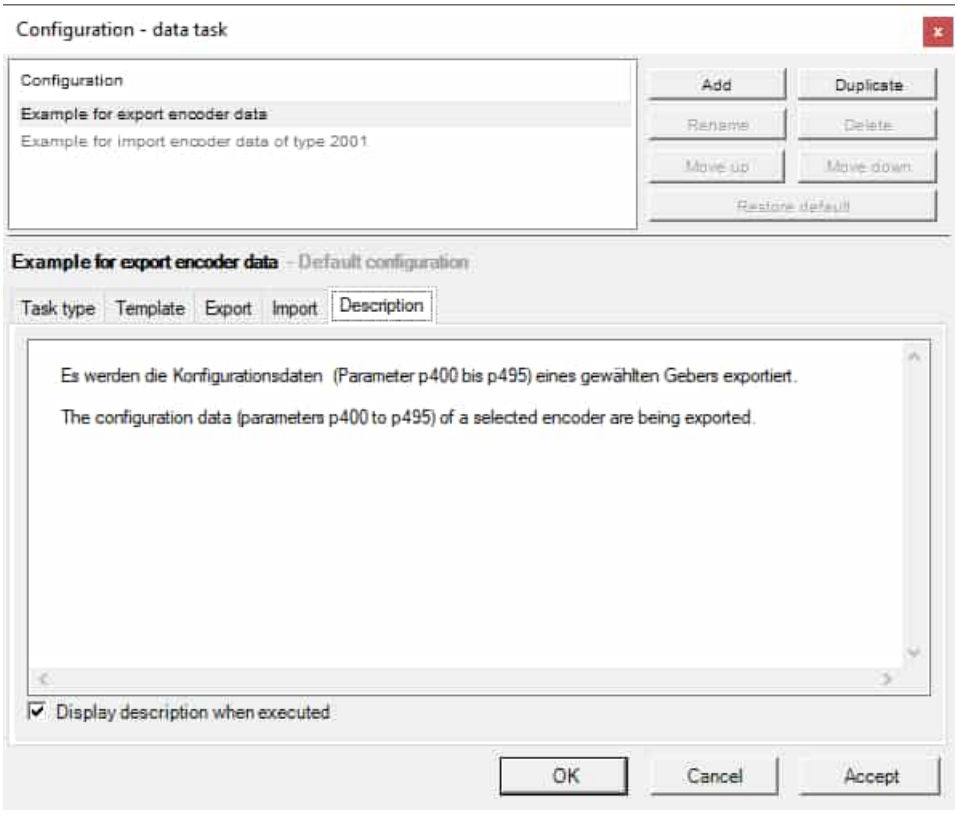
Conversion task

For a conversion task, the selection of the node type in the "Import" tab has no functional significance and cannot be selected. The node type always specifies the node on which a task is started.

Description tab

A description of the configured data task is saved in the "Description" tab.

The description is displayed in the data task wizard when the "Display description when executed" checkbox is activated.



7.2.6.3 Executing data tasks

You can execute configured data tasks using a data task wizard.

You start a data task in the selective mode of the SINUMERIK archive comparison in the Edit mode. At a marked data end node, using the shortcut menu, select "Data import with task" or "Data export with task".

The data tasks shown in gray in the task list are not permitted at the selected data end node for execution, corresponding to the configuration of the node type.

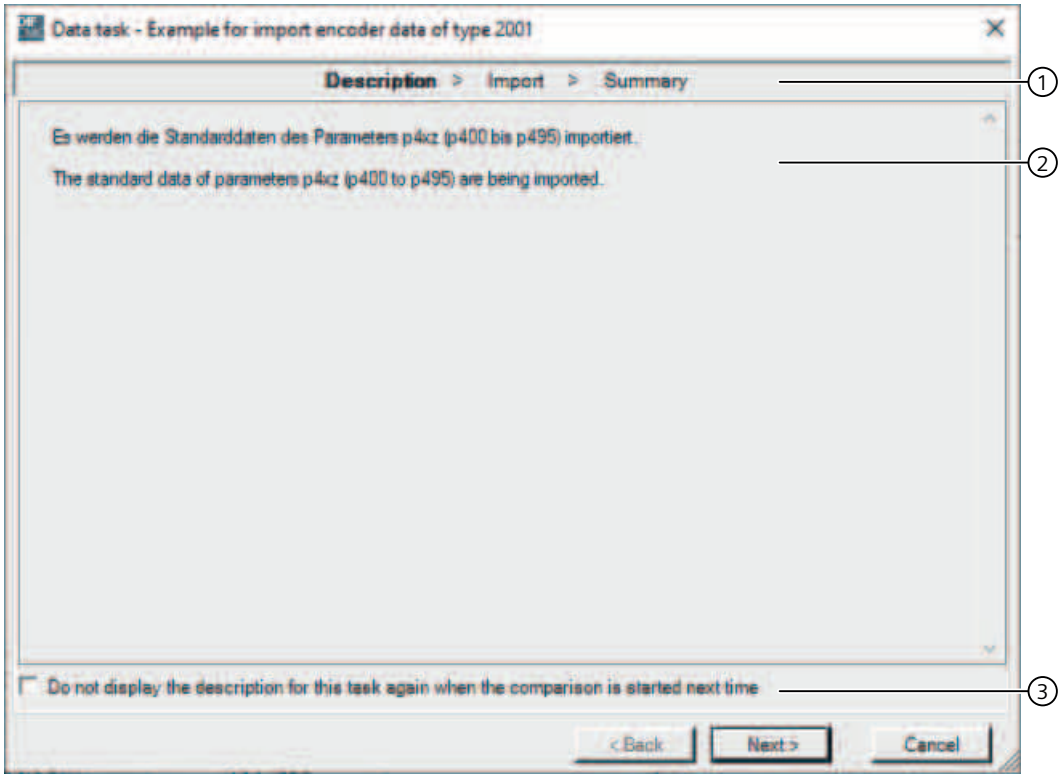
More information is provided in Section "Configuring data tasks (Page 140)".

The data task wizard starts with the 1st dialog page after clicking on the required task.

The dialog pages are subsequently described using a data import task as example.

Dialog page: Description of the data task wizard

The "Description" dialog page of the data task wizard provides the description saved in the configuration and allows users to view notes.



① **Identification of the active step**

The steps to be processed are listed in this line.

The active step is shown in **bold**.

② **Display of the description**

The description of the data task saved in the configuration is shown in this area.

③ **Deselect the dialog page**

By checking the checkbox, you can select that the wizard no longer displays this dialog page the next time this data task is started.

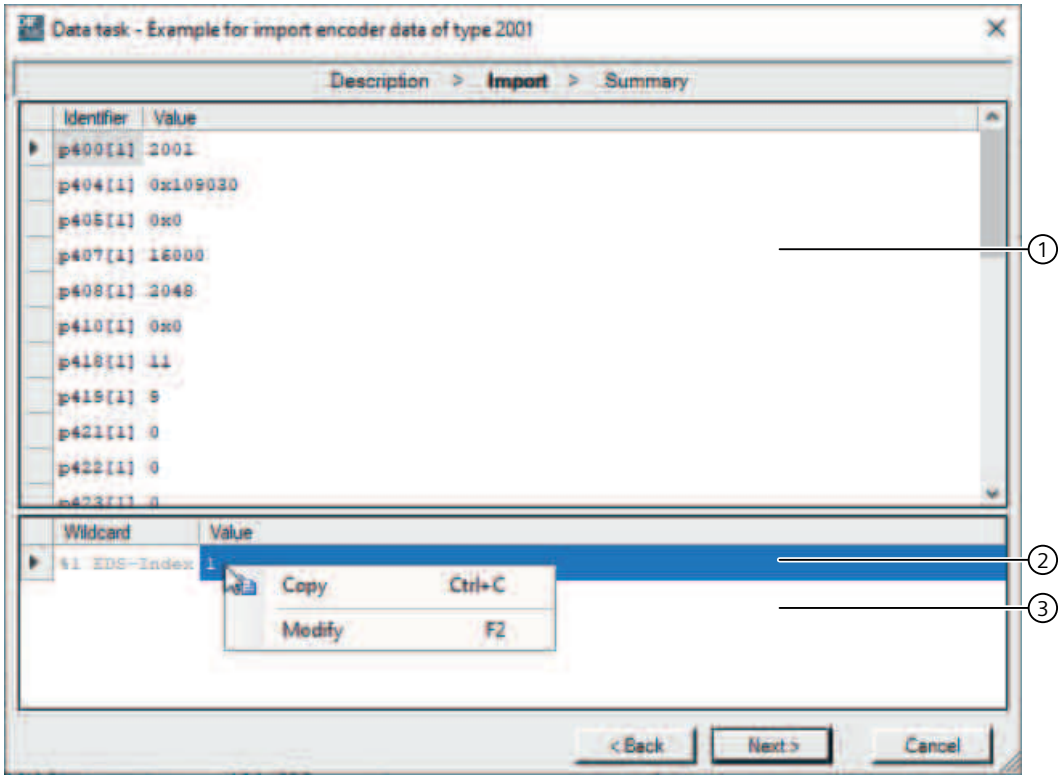
Note

If you wish to subsequently reactivate the dialog page, then this is possible via the data test configuration.

More information is provided in Section "Configuring data tasks (Page 140)".

Dialog page: Import

The "Import" dialog page allows the import values - determined by the wizard from the Excel template - to be viewed.



① **List of import designators/values**

The import identifiers from the import area of the Excel template and the associated values, which were calculated in the Excel template, are listed in this area.

② **List of wildcards/values**

The wildcards defined in the Excel template and their default values are listed.

③ **Change of the values via the shortcut menu**

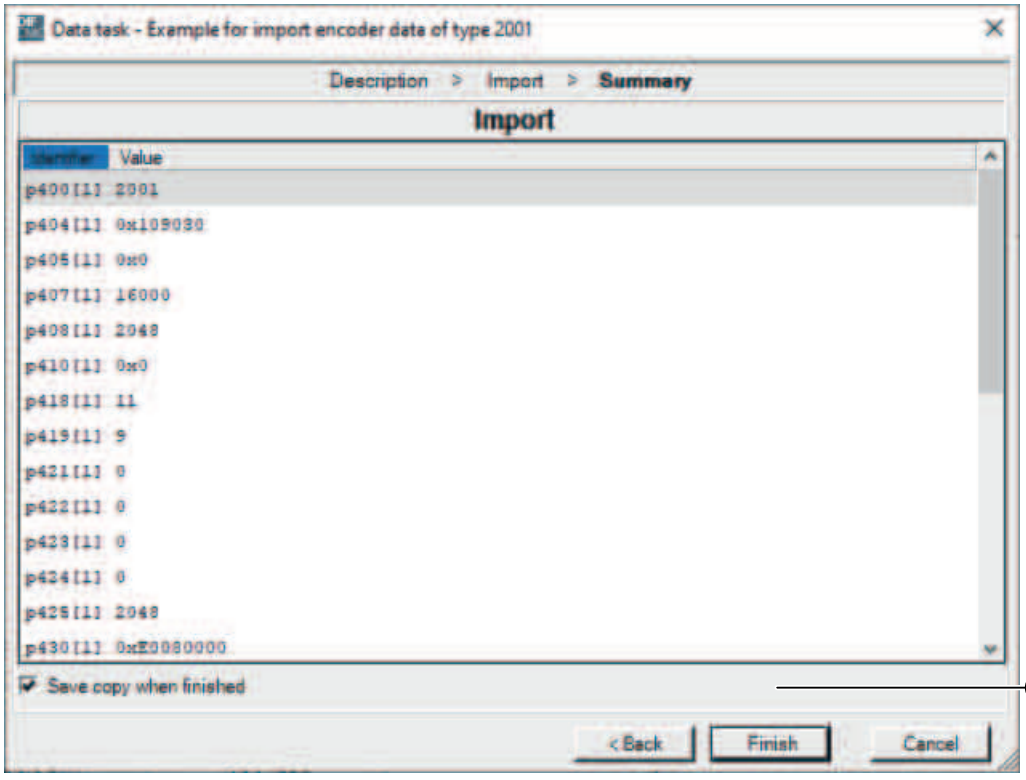
The default value saved for the wildcard can be changed using the shortcut menu.

If a wildcard, but not a default value, was defined in the Excel template, the wizard displays a question mark "?" instead of the value at ② and in the identifiers at ① - and an appropriate information text is displayed in area ②.

The next dialog is selected by clicking the "Next" button.

Dialog page: Summary

The "Summary" dialog shows the exported and imported values on a dialog and therefore provides an overview of the conversion operation.

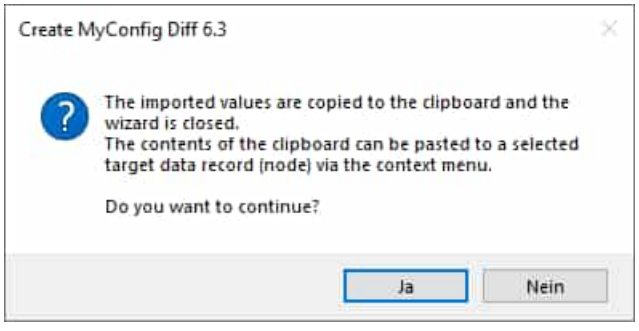


① **Saving a copy upon completion**

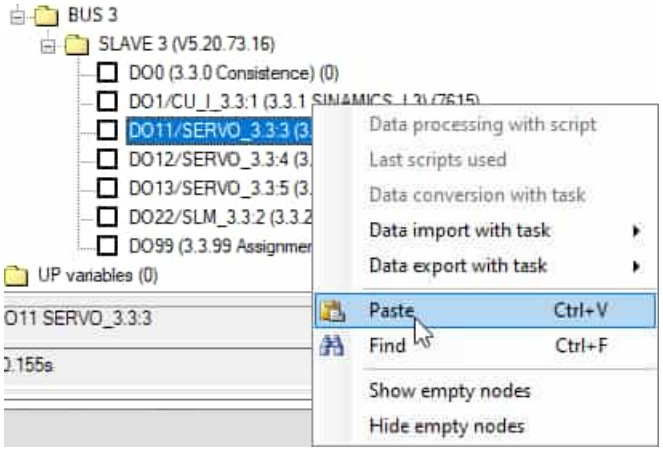
By setting a checkmark in the check box, the "Save as" dialog is called when conversion is completed; this allows the used Excel template to be saved as copy together with the values it contains.

Data conversion is completed by clicking the "Finish" button.

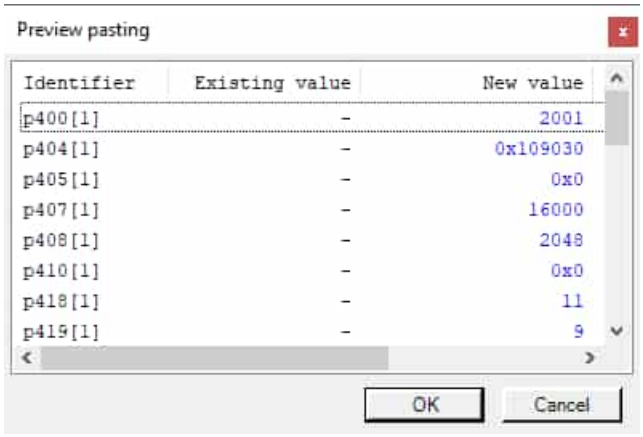
A message is then displayed that makes reference to the fact that the imported values are transferred into the clipboard. You can directly paste the imported values into a data node.



After clicking "Yes", you can paste the converted data into a data node via the shortcut menu.



Dialog box "Preview pasting" is displayed. The value changes that are obtained as a result of pasting are shown in detail.



Confirm the pasting of new values by clicking the "OK" button. Cancel the operation using the "Cancel" button.

Values that change as a result of the pasting are shown in blue.

7.2.7 Data class archives

Value change in data class archives

A data class archive can be opened and edited like a normal SINUMERIK archive in the SINUMERIK archive comparison. A feature of the "*.ard" archive is that **only** values in **data classes** can be changed. New identifiers cannot be added nor can existing identifiers be deleted.

Opening a data class archive

Proceed as follows to make changes to a data class archive:

1. Select via quick start or the "File" > "SINUMERIK archive comparison" menu.
2. Open the desired data class archive via "File" > "Open".
   Following the path and the designation of the data class archive, the data classes contained in the archive are shown in square brackets.

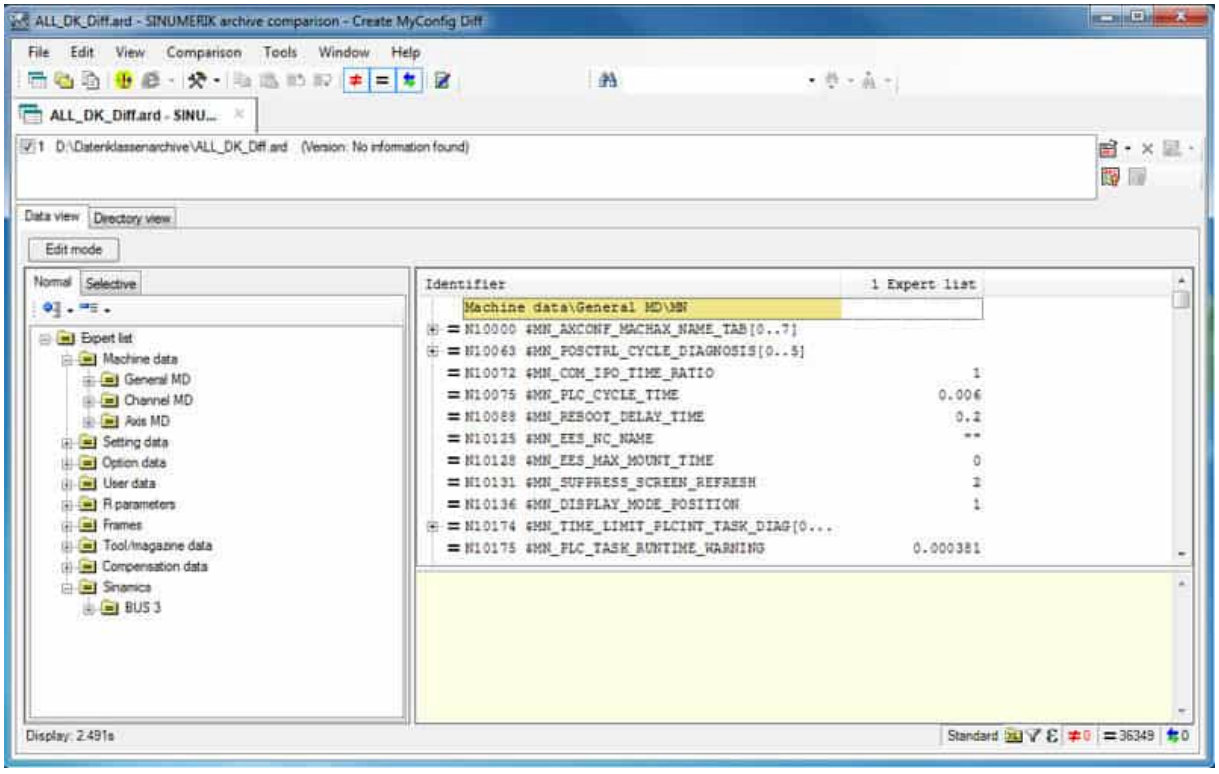


Figure 7-12 Opening a data class archive

3. In the opened data class archive, you can change the associated values in the "Data view" tab > "Normal" tab or "Selective" tab in activated **Edit mode**.

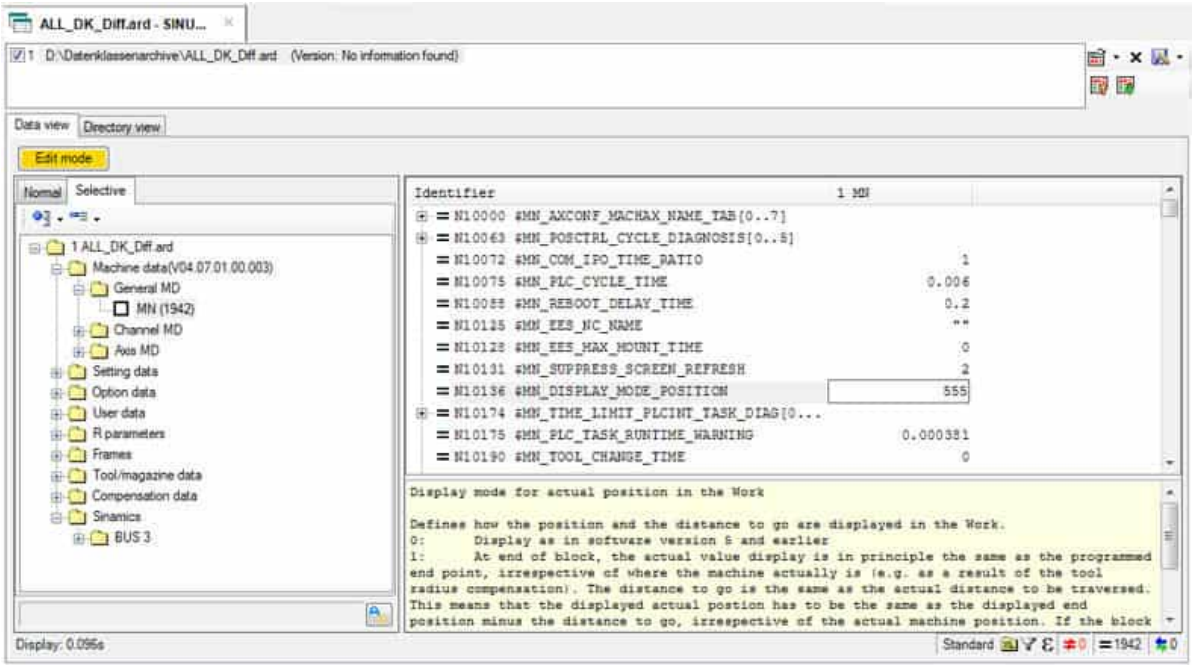


Figure 7-13 Data class archive: Change value

4. Values can also copied from an identifier and pasted in a different identifier. After pasting, the "Preview pasting" window opens to show the old value and the new value.

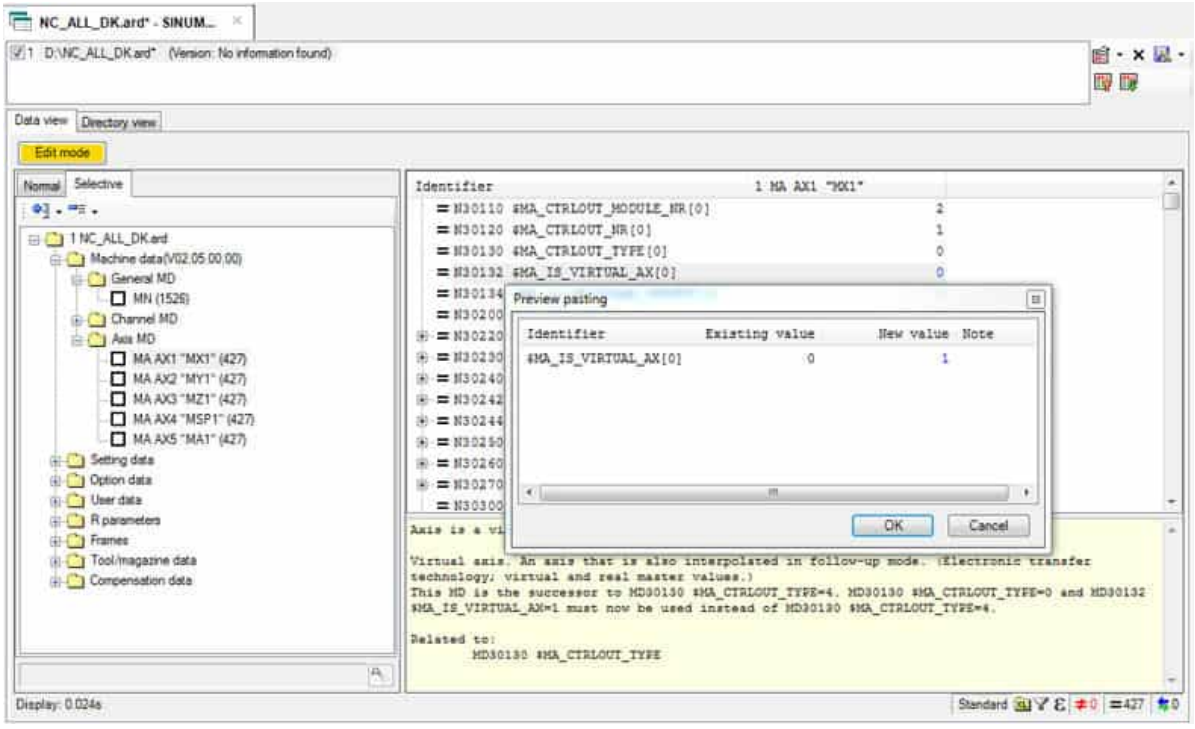


Figure 7-14 Data class archive: Copying and pasting values

7.2.8 CMC scripts in Diff

7.2.8.1 Introduction

Analysis and manipulations of archives via CMC scripts in Diff

Diff supports the advanced analysis and manipulation of archives via CMC script. You can implement, for example, complex data export operations using the script functions to write to files. A command line interface for executing the CMC scripts in Diff is provided for automation tasks.

Note

License

You require a license to use CMC scripts in Diff.

The function as extension "DIFF Scripting" is under license.

More information

In Diff you can apply CMC scripts on SINUMERIK archives.

This allows you to make complex controlled data exports from archives, without having to create a CMC package. You can also manipulate the data archive.

You can start the scripts from the Diff interface - or using a command line call.

Note

Exceptions in CMC scripts in Diff

"CMC scripts in Diff" are subject to certain restrictions.

Details on this are provided in the section or in the online help "Create MyConfig CMC script (Page 29)".

More information on CMC scripts

More information about "CMC scripts" is provided in Section "Create MyConfig CMC script (Page 29)" in the Operating Manual or in the online help with the same name.

7.2.8.2 Data accesses / sections / identifiers

You must always use fully qualified sections to be able to read and write access archived data, e.g. `NC[C1], PS[B3_S3_PS3]`.

Use simple identifiers under fully qualified sections.

You only use fully qualified identifiers for read access operations.

Example

```
NC[C1] ;Fully qualified section
$MC_CHAN_NAME=InputText("Enter the required channel names:","");
simple identifier
Up.KName1=$MC_CHAN_NAME ;simple identifier
MSG("Channel name for channel 1 was set to $(Up.KName1).")
Up.KName2=NC[C2].$MC_CHAN_NAME ;fully qualified identifier
MSG("Channel name for channel 2 is $(Up.KName2).")
```

Note

No access to display data (HMI) in the archive

It is not possible to access display data (HMI) in the archive from CMC scripts in Diff.

Note

Write access to archive data

Activate the "Edit mode" in Diffto write archive data to the CMC script.

Check system variable `Up.$Script.EditMode` to determine if the "Edit mode" is activated

7.2.8.3 System variables

More information about system variables of CMC scripts in Diff is provided in Section "CMC script" > "System variables - CMC scripts in Diff" (Page 34) or in the online help "Create MyConfig CMC scripts".

7.2.8.4 Calling scripts from the interface

Calling scripts using the list of opened archives

To call a script in Diff, right-click the required archive in the list of opened comparisons.

In the context menu, select either "Data processing with scripts" or "Last scripts used".

For read-only access to the archive, the Edit mode must not be switched on. Activate the Edit mode when writing to the archive or making changes in it.

A CMC script is executed by Shield.

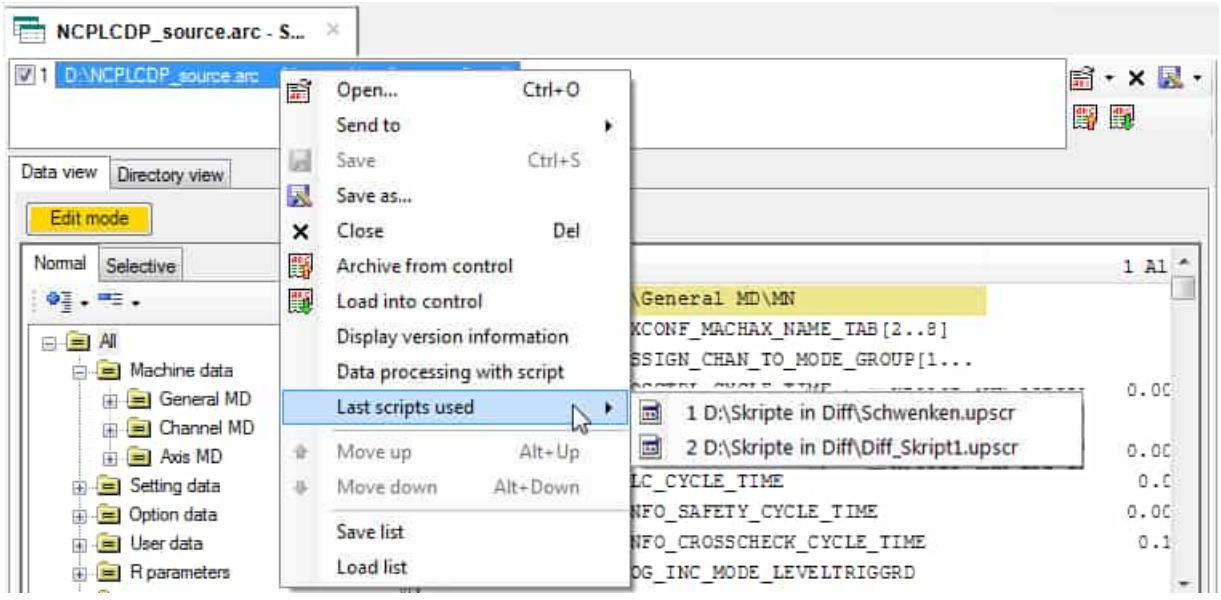


Figure 7-15 Calling a CMC script via an archive list

Alternative script call in "Data view" tab > "Selective" tab

Call a script via the root folder of the archive with context menu in the "Selective" tab.

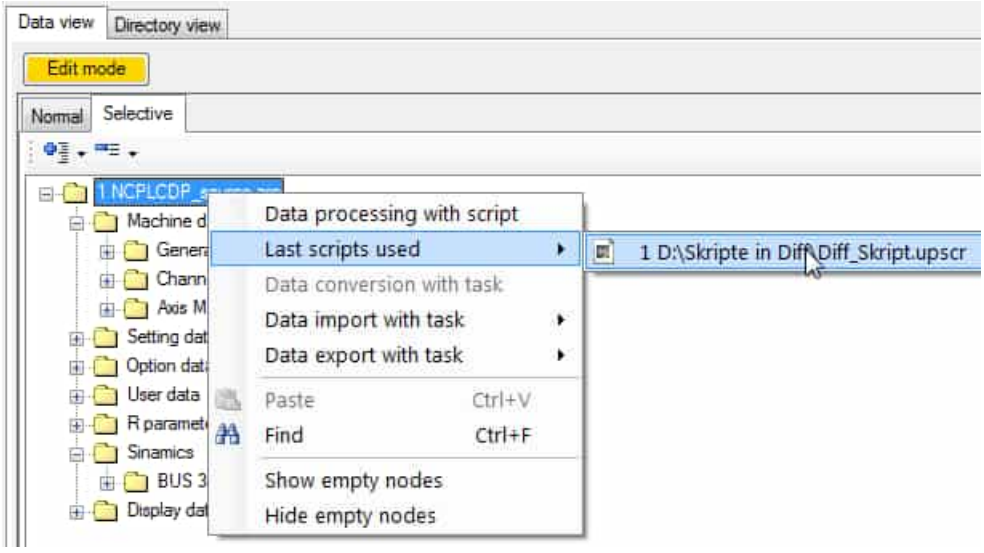


Figure 7-16 Calling a CMC script in the selective mode

7.2.8.5 Calling scripts from command line

CMC scripts can also be started without the graphic user interface of Diff from the command line.

The syntax is described in Section Command line call (Page 200).

7.3 Folder comparison

7.3.1 General information

In a Folder comparison of Diff, you can compare 2 comparison objects with each other, insert or delete files or folders and resave the modified objects. The comparison result is compared on the folder and file levels and then shown. This means that the contents of the files are compared optionally, but not displayed.

You can compare and edit the following objects using the Folder comparison:

- Folders/directories
- SINUMERIK ONE archives in the DSF format
- ZIP archives
- TGZ archives
- TAR archives

Because unpacked archives often represent a structure consisting of folders and files, similar rules apply to comparing and editing.

In the folder comparison, files with the same name are compared for size, date, version, and binary content.

The comparison result is prepared both qualitatively and quantitatively.

From the comparison objects, folders and files can be copied to the clipboard and to other applications (Expert, Windows Explorer, etc.).

7.3.2 Overview of user interface

7.3.2.1 Main menus

The main menus in Diff are identical. The submenus are adapted dynamically to the associated SINUMERIK archive comparison, Folder comparison or File comparison component.

The following main menus and submenus are available in the Folder comparison:

- File
- Edit
- View
- Comparison
- Tools
- Window
- Help

File

Table 7-10 Diff - Folder comparison - File main menu

File			
	SINUMERIK archive comparison		Opens a new SINUMERIK archive comparison
	Folder comparison		Opens a new Folder comparison
	File comparison		Opens a new File comparison
	Open folder	Left-hand side... Right-hand side...	Opens a folder on the left-hand or right-hand side in the Folder comparison.
	Open archive...	Left-hand side... Right-hand side...	Opens an existing archive on the left-hand or right-hand side.
	Save	Left-hand side... Right-hand side...	Saves a selected archive either on the left-hand or right-hand side in the Folder comparison.
	Save as	Left-hand side... Right-hand side...	Saves an archive opened on the left-hand or right-hand side in the Folder comparison with a different name or location.
	Archive from controller	Left-hand side... Right-hand side...	Loads an archive from the control into the left-hand or right-hand side in the Folder comparison.
	Load into controller	Left-hand side... Right-hand side...	Loads an archive on the left-hand or right-hand side directly into the control.
	Exit	<Alt+F4>	Exits the Diff program.
	Close	Left-hand side... Right-hand side...	Closes the selected archive on the left-hand or right-hand side in the Folder comparison.

Edit

Table 7-11 Diff - Folder comparison - Edit main menu

Edit			
	Cut	<Ctrl+X>	First activate the "Edit mode" button in order to remove selected objects and copy to the clipboard.
	Copy	<Ctrl+C>	Copies the currently selected objects to the clipboard.
	Paste	<Ctrl+V>	First activate the "Edit mode" button in order to paste objects from the clipboard to the selected position.

Edit			
	Delete	<Del>	First activate the "Edit mode" button in order to delete selected objects.
	Select All	<Ctrl+A>	Selects all lines of a folder within a comparison.
	Modify	<F2>	Changes or renames folders or files.
	Go to next difference	<Ctrl+N>	Displays the differences in direction, forward or backward.
	Go to the previous difference	<Ctrl+B>	

View

Table 7-12 Diff - Folder comparison - View main menu

View			
	View only left occurrences	<Ctrl+Shift+L>	Displays those objects available only in the **left-hand** comparison object. You can activate or deactivate the display.
	Different	<Ctrl+Shift+D>	Displays those objects **unequal** in both comparison objects. You can activate or deactivate the display.
	Equal	<Ctrl+Shift+E>	Displays those objects **equal** in both comparison objects. You can activate or deactivate the display.
	View only right occurrences	<Ctrl+Shift+R>	Displays those objects available only in the **right-hand** comparison object. You can activate or deactivate the display.
	Column layout	Same order	Specifies the sequence of the columns in the comparison result list.
		Mirrored	
	Representation	Tree	The display in the Folder comparison can be switched between the tree structure and the list structure.
		List	
	Toolbars	Standard	Like the search, the standard toolbar can be shown or hidden. The displays are activated as default.
		Search	

Comparison

Table 7-13 Diff - Folder comparison - Comparison main menu

Comparison				
	Refresh	<Ctrl+D>	The "Refresh comparison" button is color-highlighted when the displayed comparison result is not up-to-date. Start the update of a comparison.	
	HTML protocol		HTML protocol	Clicking the HTML protocol button displays in the standard browser (Internet Explorer) the comparison result of the node selected in the data structure. You can save the result either as HTML or XML protocol.
			HTML protocol data fields as displayed	
			XML protocol	
			XML protocol data fields as displayed	
	Edit configuration...	<Ctrl+M>	The "Edit configuration..." icon changes a comparison configuration or selects a different comparison configuration. More information is provided in Section "Configuration - folder comparison (Page 169)".	
	Configuration selection		Standard	Switches the comparison configuration directly without dialog.
			SINUMERIK archives	

Tools

Table 7-14 Diff - Folder comparison - Tools main menu

Tools			
	Options...	Opens the "Options" dialog with the tabs described below. Options can be activated or deactivated for representation/execution of the Diff components. All options are **activated** as **default**. The Diff coloration and the language setting can be changed user-defined.	
		General	The following settings can be activated or deactivated: • Recompare objects automatically when the comparison configuration changes • Display quick start when Diff starts
		Folder comparison	**Automatic selection of comparison configuration** The following settings can be activated or deactivated: • Select the "SINUMERIK archives" configuration for the SINUMERIK archive comparison • Message for automatic configuration change • Compare NCU folders with different names in the archive as having the same name. (NCU_* == NCU_*) Restart required
		File comparison	**Automatic selection of comparison configuration** The following settings can be activated or deactivated: • Identify the file type and select the configuration • Message for automatic configuration change **File identification configuration...** Clicking this button opens the "Configuration - file identification" dialog. You will find more information about this in section "Configuring

Tools			
			automatic file identification (Page 195)."
		SINUMERIK archive comparison	**Parameter** The following setting can be activated or deactivated: • Display, copy and export with parameter short name
		Colors	You can change the color setting for the colored line identification of the comparison results. This option is useful, for example, for a red-green sight impediment. You can restore the default setting: • Unequal - Red • Equal - Black • Right - Blue • Left - LimeGreen
		Language	Language selection of the Diff interface. Available languages are: • German • English

Tools			
	Data comparison	Data comparison configuration...	Opens the "Configuration - NC data comparison" dialog. Information can be found in Section "Configuration - NC data comparison (Page 127)".
		Configuration data task	This selection opens the "Configuration - data task" dialog. Information can be found in Section "Data tasks configuration (Page 140)".
	Folder comparison	Folder comparison configuration	Opens the "Configuration - folder comparison" dialog. Information can be found in Section "Configuration - folder comparison (Page 169)".
	File comparison	File comparison configuration...	Opens the "Configuration - file comparison" dialog. Information can be found in Section "Configuration - file comparison (Page 189)".
		File identification configuration...	Opens the "Configuration - file identification" dialog. Information can be found in the "Automatic file identification configuration (Page 195)" section.
	Export configurations		The following configuration types and the associated files can be selected or deselected in the "Export configurations" dialog: • Archive/folder comparison configurations • File comparison configurations • Data comparison configurations • File identification configurations • Data tasks Only the selected configurations are exported.
	Import configurations		Configurations are imported via the path selection.

Window

Table 7-15 Diff - Folder comparison - Window main menu

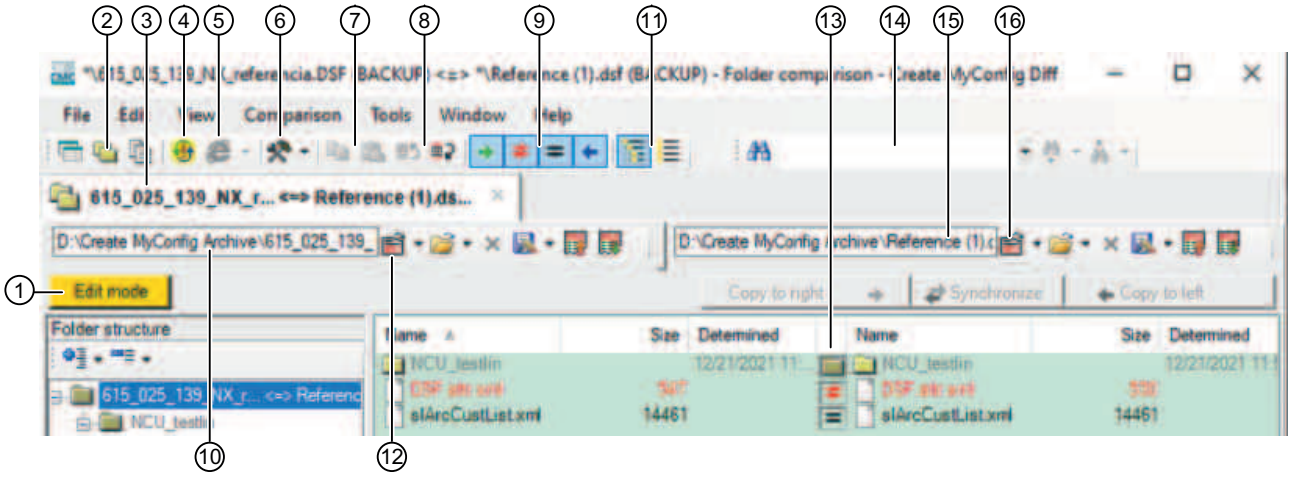
Window		
	New horizontal register group	Displays the selected component in the new horizontal register group.
	New vertical register group	Displays the selected component in the new vertical register group.
	Move to previous register group	Moves the selected component to the previous register group.
	Move to next register group	Moves the selected component to the next register group.
	Folder comparison	Displays the opened folder comparison, left-hand side ⇔ right-hand side.

Help

Table 7-16 Diff - Folder comparison - Help main menu

Help			
	Content...	<F1>	Display of the online help for the **Diff** program. The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	CMC script...		Display of the Online Help "CMC script" The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	FAQs...		Display of the Online Help "FAQs" The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	Display quick start...		You open a SINUMERIK archive comparison, folder comparison, file comparison or the online help via the "Quick start" window.
	About...		Contains information on the version and copyright for Create MyConfig Diff.

7.3.2.2 Folder comparison - toolbar



① Buttons for "Edit mode" - "Synchronize" - "Copy to right" / "Copy to left"

② New folder comparison
Calling a folder comparison.

③ Toggling between opened comparisons (tab)

④ Refresh comparison

⑤ Display/save HTML/XML protocol

⑥ Configure comparison

⑦ Copy / Paste

⑧
Go to next/previous difference

⑨ Button group comparison results

⑩ / ⑮ Path of the opened comparison object (**left** side / **right** side)

⑪ Switchover between tree and list display.

⑫ / ⑯ Functions - selecting the comparison object on the **left** / **right** side

⑬ Column with the comparison result icons

⑭ Buttons/input for searching

Description of buttons/fields of the toolbar

The buttons of the toolbar are described in more detail below.

① Edit mode - Synchronize - Copy



Use this button to switch the Edit mode on or off.

When the Edit mode is activated, using the shortcut menu, the following edit functions can be applied:

You can cut, copy, paste, delete and rename files and folders. You can also create new files and folders.

The changes are saved in an archive using the floppy disk icon.

Note

Changes become effective immediately

Changes in a folder become effective immediately in the original folder and **cannot** be saved again.

As a consequence, the floppy disk icon remains deactivated.



Use this button to synchronize the content of the left comparison object with the content of the right comparison object.

You can influence which files and folders should be adapted by changing the comparison icons with a left click or from the shortcut menu.



With the "Copy to right" button, the selected files and folders of the left-hand comparison object are inserted in the right-hand comparison object.



With the "Copy to left" button, the selected files and folders of the right-hand comparison object are inserted in the left-hand comparison object.

③ Switchover between opened comparisons

In the opened tabs, similar to the Windows task bar, you can toggle between the opened comparisons with a mouse click or by simultaneously pressing "Ctrl" and "Tab".

④ Refresh comparison

The refresh button is always shown in color if the displayed comparison result is not a current result. If this is the case, restart the comparison using this button.

⑤Display/save HTML/XML protocol

This button displays the comparison result of the node selected in the folder structure in the standard browser (Internet Explorer).

You can save the result either as HTML or XML protocol.

The comparison result for the selected directory is output completely or, under "...as displayed", an output depending on the settings made under ⑨.

⑥ Configure comparison

Using this button, a comparison configuration is changed or a different comparison configuration is selected.

More information can be found in Section "Configuration (Page 169)".

⑦ Copy / Paste

Buttons for copying and pasting selected folders or files of the left-hand or right-hand comparison object. The buttons are only activated in Edit mode.

More information can be found in Section "Copying from the folder comparison (Page 173)".

⑨ Button group comparison results

Use this button group to show or hide the corresponding lines of the comparison result.

⑪ Switchover between tree and list view

Improved synchronization functionality is obtained by being able to toggle.

⑫ / ⑯ Selection of the left-hand / right-hand comparison object

 Select archive for comparison

 Select folder for comparison

 Selection from the objects that were last selected

 Load archive from controller

 Import archive into the control

⑬ Column with the comparison result icons

For separating the left-hand and right-hand comparison objects.

You can change the icons for the purpose of synchronization when the Edit mode ① is activated (left click, shortcut menu).

⑭ Buttons/input for searching

This group contains:

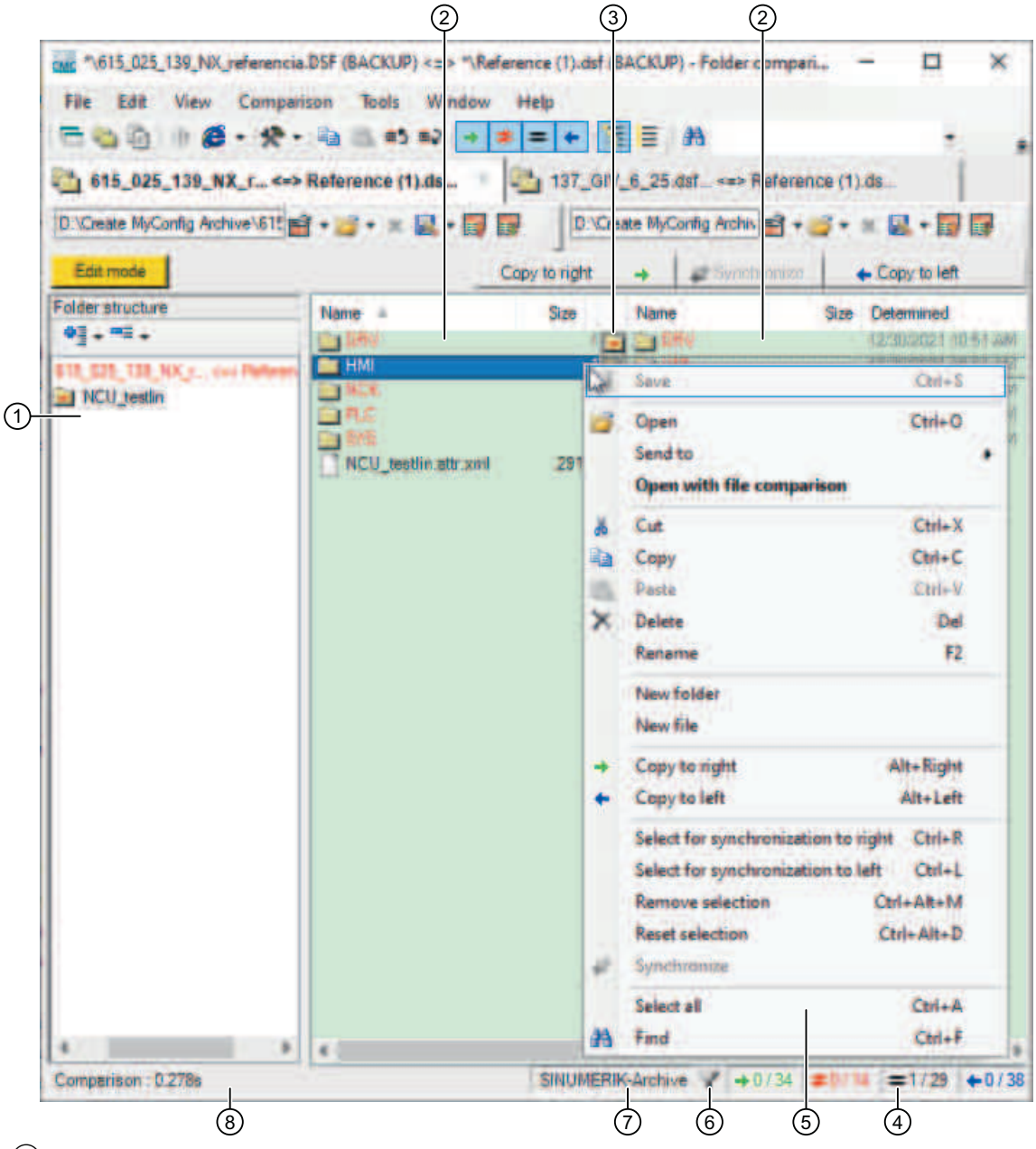
- A button for opening a dialog, in which you can define the search options
- An input field for searching for character strings in selected nodes
- Buttons for searching upward and downward

 Search down

 Search up

7.3.2.3 Folder comparison - overview window

Below is an overview of the functions in the "Folder comparison" window.



① Folder structure - Representation of the folders with the associated comparison objects in the tree structure
② Colored comparison of the comparison objects
③ Column of the comparison result icons
④ Assignment of the comparison results icon/value (status line bottom right)
⑤ Shortcut menu
⑥ Configure comparison

⑦ Display of the current comparison configuration (status bar)
⑧ Status bar

Description of the areas/functions in the Folder comparison window

The individual functions of the folder comparison are described in more detail below:

① Folder structure - Representation of the folders with the associated comparison objects in the tree structure

In this area, the folders of the comparison objects are shown in a tree structure arranged one on top of the other. The comparison result is visualized using special folder icons.

② Colored comparison of the comparison objects

The content of the folder selected in the ① area is displayed next to each other for both comparison objects (②). The center column ③ with the comparison result icons also serves as separation between the left-hand and right-hand comparison objects.

The comparison results are color-coded to make it easier to interpret the results:

- Objects that are only available on the left-hand side are shown in **green**.
- Unequal objects or properties available on both sides are shown in **red**.
- Equal objects or properties available on both sides are shown in **black**.
- Objects that are only available on the right-hand side are shown in **blue**.

The contents of the column can be sorted in ascending or descending order by clicking the column header.

Note

The contents of the files are only compared in the binary notation if you preselected this in the comparison configuration. More information is provided in the "Configuration (Page 169)".

The date for the comparison is not relevant for folders.

Note

PLC blocks identification

If PLC blocks are marked **yellow**, they are **Safety Integrated plus** blocks.

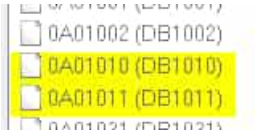


Figure 7-17 PLC blocks identification - Safety Integrated plus

③ Column of the comparison result icons

Column with the comparison result icons, to separate the left-hand and right-hand comparison objects.

When the Edit mode is activated, you can change the icons for synchronization purposes by left-clicking or via the shortcut menu.

Significance of the comparison icons:

- A green arrow indicates that the folder is only available in the left-hand comparison object.
- ≠ The unequal character indicates that the contents of the folder are not identical.
- = The equal character indicates that the contents of both folders are identical.
- A blue arrow indicates that the folder is only available in the right-hand comparison object.

④ Assignment of the comparison results icon/value (status bar bottom right)



The value in front of the slash specifies how many of these files can be seen in the selected folder.

The value after the slash shows the total number of these files that are contained in both comparison objects.

Example:



Specifies that there are "8" unequal files from a total of "14" in the currently selected folder.

⑤ Special features shortcut menu

The "Open", "Send to" and "Copy" entries can be used to view and extract files from archives.

The entry "Open with file comparison" or a double-click opens the files contained in the selected line in the data comparison of Diff.

The other entries can be used to edit a selected file or a selected folder.

⑥ Configure comparison

A filter is used in the comparison configuration. More information can be found in Section "Configuration (Page 169)".

The comparison configuration is opened with a double-click.

Note

Using inclusion and exclusion filters, it is possible that not all of the files of the folder or archive are displayed any longer.

⑦ Display of the current comparison configuration (status bar)

The comparison configuration that is being used in the current comparison is displayed in this field. More information can be found in Section "Configuration (Page 169)".

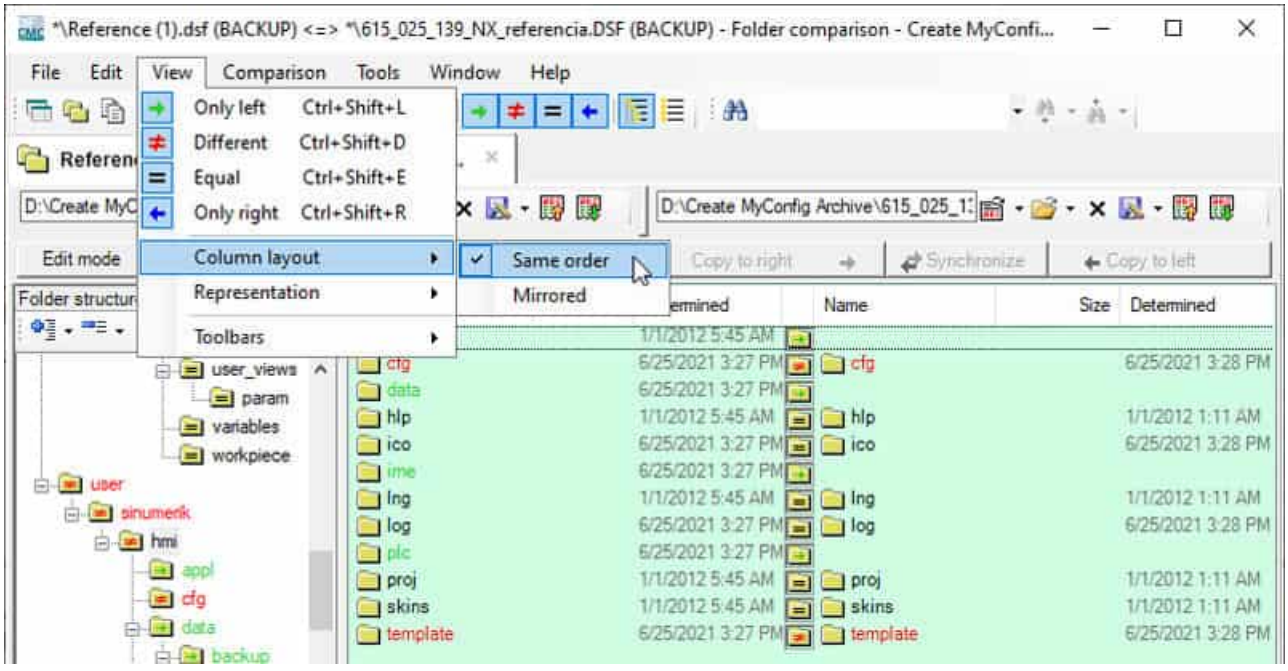
The comparison configuration is opened with a double-click.

7.3.3 Changing the column layout

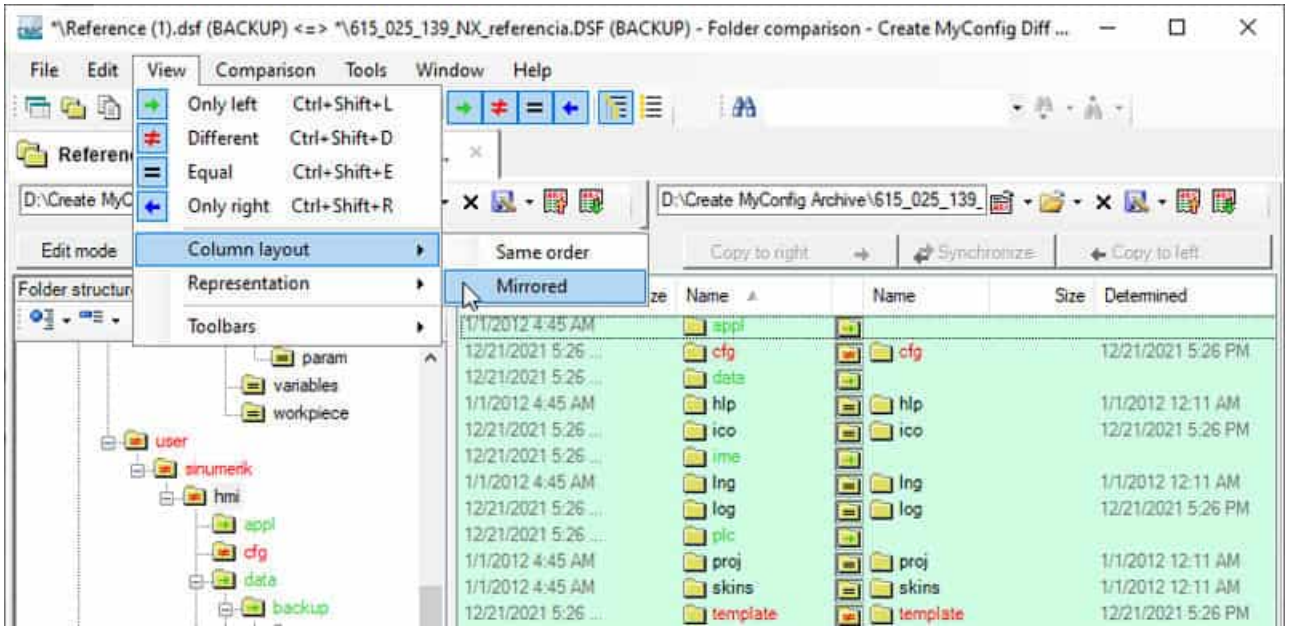
Changing the column layout for folder comparison

You can change the column arrangement when comparing folders in the menu "View" > "Column layout" > "Same order" or "Mirrored".

Same order



Mirrored



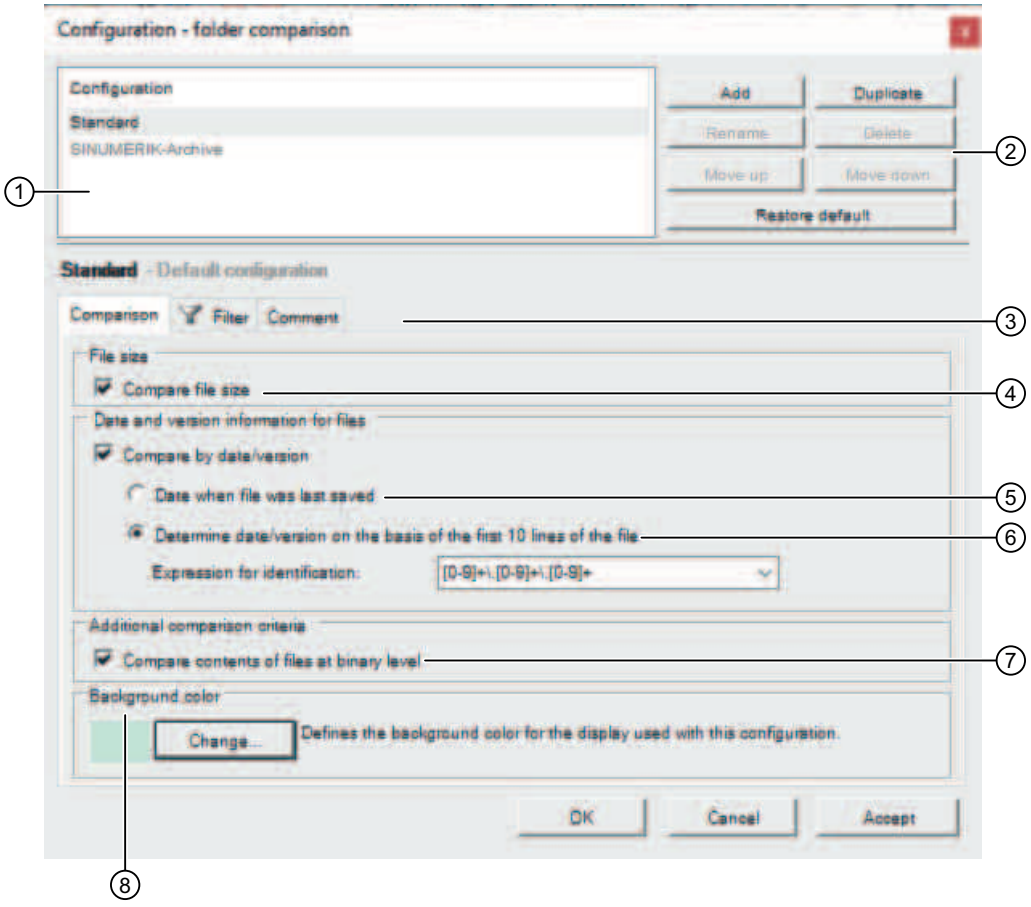
7.3.4 Configuration

Folder comparison

Clicking the "Configure comparison" button calls the "Configuration - folder comparison" dialog. You can manage various comparison configurations in the dialog. You can create, edit and delete comparison configurations.

All criteria to be considered for a folder comparison are defined in a comparison configuration. You can also export or import configurations, for instance, to make them available for other users.

More information is provided in Section "Exporting and importing configurations (Page 198)".



① List of the comparison configurations
② Buttons for managing the configurations
③ Comparison tab / Filter / Comment
④ Compare file size
⑤ "Date when file was last saved" option button
⑥ "Date/version" option button / "Identification regular expression" drop-down list
⑦ "Additional comparison criteria" checkbox
⑧ Background color

① List of the comparison configurations

The existing comparison configurations are listed in this area of the configuration dialog and can be selected by clicking them.

Configurations shown in gray are user-specified configurations; these can be changed, but not deleted - and can be restored again at any time.

② Buttons for configuration management

Configurations can be managed using these buttons.

Using "Add", a new configuration is generated, which can also be adapted.

A copy of the selected comparison configuration is created by clicking "Duplicate".

With "Restore default", the changes to the specified comparison configuration are undone.

Note

The "Restore default" button is only active if a user-specified configuration has been selected, the contents of which are no longer the same as the default setting.

In addition to restoring the default setting, this button can also be used to check whether the contents are identical with those of the default setting.

③ **Comparison tab / Filter / Comment**

The properties that belong to the selected configuration are stored in this area of the "Configuration - folder comparison"; these properties include comparison criteria, filter criteria and a comment.

Comparison tab

④ **Compare file size**

For activated size comparison, the size of the files and folders is considered.

If you do not activate the option checkbox, a check is only made whether files and folders exist in both comparison objects.

This option is activated as standard.

⑤ **Date when file was last saved option button**

When the date/version comparison is activated, then for this selection, Diff also takes into account the date when the file was last saved - and displays this in either black or red.

The date is not relevant for folders.

⑥**Date field/version/drop-down list of regular identification expressions option button**

SINUMERIK archives can contain cycles.

The cycle version number is specified in the file. When making a comparison, the version number provides more information than the date and can be used for the comparison.

In order to be able to compare the version numbers, the files in the first 10 lines are checked for the character string, which is then described using a regular identification expression.

A regular expression that is generally used is already selected in the specified comparison configuration "SINUMERIK archives".

You can also create user-defined regular expressions in the drop-down list.

An introduction to regular expressions is provided in Section "Using regular expressions in Diff (Page 206)".

⑦ **"Additional comparison criteria" checkbox**

The content of the files is only compared byte-by-byte when you select the additional comparison criterion "Compare binary content of the files".

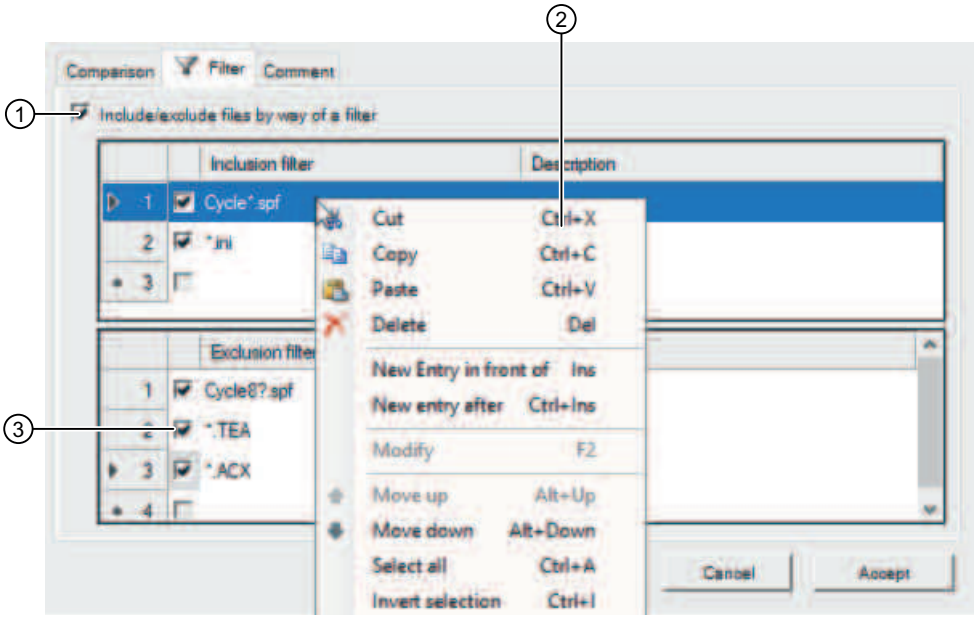
Otherwise, only the size is compared and possibly the date or the version.

⑧ **Background color**

So that the user can immediately recognize how to select certain comparison configurations, you can define a separate background color for each comparison configuration.

Filter tab

Using filters, for the comparison, you can display or hide files based on screen forms.



① The functionality of the inclusion and exclusion filter is activated via the "Include/exclude files by way of a filter" checkbox.

NOTE:

Filter icon is shown in the status bar if this function is activated.

② You can create, edit and delete filter entries using the shortcut menu.

③ You can individually activate or deactivate each filter entry.

Table 7-17 Examples of filters:

***.ini**	The "*" (asterisk) expression represents any number of any characters. If the entry is defined in the inclusion filter, only file names with the file extension "ini" are displayed, and all others are hidden. If "*.ini" is present in the exclusion filter, all file names with the "ini" extension are not displayed.
cycle8?.spf	The question mark represents precisely one character (any character). If you have defined the entry in the inclusion filter, files with the "cycle8" name plus any arbitrary character and the "spf" file extension are displayed, and all others are hidden. File names with the designation "cycle8?.spf" can therefore be called as follows: cycle80.spf, cycle81.spf, cycle8a.spf..., etc.

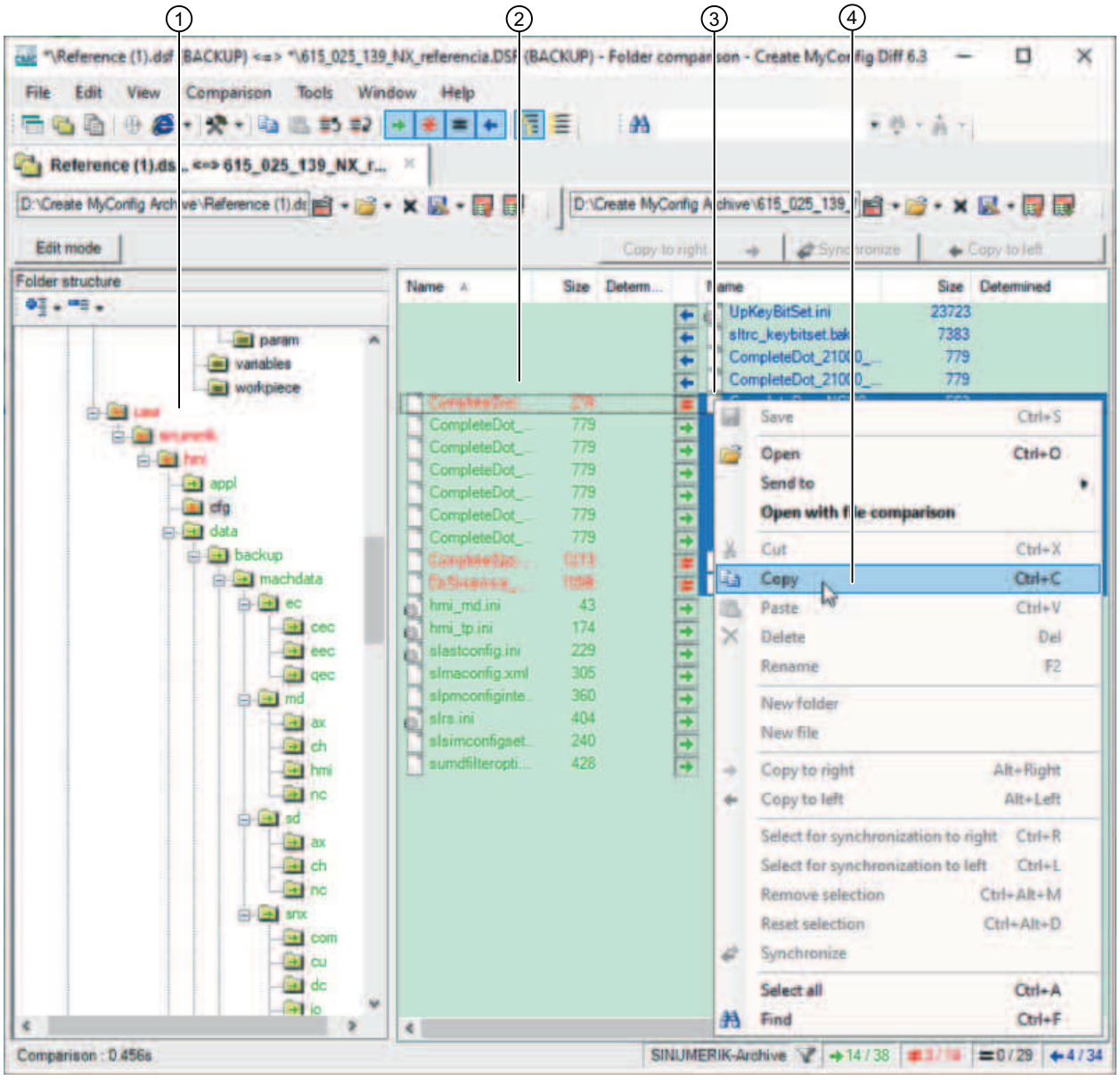
Comment tab

You can save any comments for the selected comparison configuration under the "Comment" tab.

7.3.5 Copying from the folder comparison

Files, folders or complete folder structures can be copied from the comparison objects of the folder comparison to the clipboard and then, e.g. in Expert, inserted in a component.

This means that it is possible to extract files from SINUMERIK, ZIP, TAR or TGZ archives, without you having to use software that is not part of the Create MyConfig landscape.



In the selected folder ①, in the list of folders and files contained in it ②, the objects to be copied ③ are selected using standard Windows tools (<Ctrl + mouse click>, <Shift + mouse click> etc.).

Note

It should be noted that the objects to be copied can only be selected on the left-hand side or only on the right-hand side of the comparison objects.

Copy using the usual Windows tools, e.g. shortcut menu ④, button or <Ctrl+C>.

7.3.6 Save archive as user or SDB archive

The following archives with different properties are available for the SINUMERIK:

- SINUMERIK archive DSF
- Commissioning archive (Commissioning archive)
- USER archive
- SDB archive (PLC hardware upgrade archive, PLC HW Config archive)

Saving various archive types

The file type is selected and saved in the folder comparison in dialog "Save archive".

The following archives can be generated:

In addition to the SINUMERIK ONE archive DSF, you can create the following archives in the Diff folder comparison:

- **SINUMERIK archive DSF**
  In SINUMERIK ONE, the "Backup" and "Setup" DSF archives are used and supported over the complete workflow.
  For CMC, the "Backup" DSF archive is relevant. Control data is backed up and can be restored again.
  More information on the DSF archive is provided in the System Manual SINUMERIK ONE Create MyVirtual Machine (https://support.industry.siemens.com/cs/ww/en/view/109801361).
- **Commissioning archive (SINUMERIK 840D)**
  - "Standard SINUMERIK archive", just the same as it is also created in SINUMERIK Operate, it can include all data areas and control commands for a memory reset, loads standard data etc.

- **USER archive (SINUMERIK 840D)**
  - Only contains the data areas (folders) HMI, PLC and NC; only these data areas are backed up.
  - Does not include any drive and NC machine data - and no PLC-SDB blocks (HW configuration).
  - When loading, the memory is not reset, standard data is not loaded and no NCK reset is executed.
  - It is only possible to read in USER archives via SINUMERIK Operate in the system data nodes archive area.
- **SDB archive (SINUMERIK 840D)**
  - PLC hardware upgrade archive, just as SINUMERIK Operate also generates
  - Only contains the data areas (folders) PLC and CP - only these data areas are backed up.
  - Only the system data blocks SDB0 to SDBn (0B00000 to 0BFFFFF) are saved in the PLC folder.
  - When loading, the PLC memory is not reset, i.e. user data and programs are kept.

7.4 File comparison

7.4.1 General information

In a file comparison using Diff, you can compare the contents of two files line-by-line.

The comparison result is processed from qualitative and quantitative perspectives.
Contents can be copied from the files line-by-line to the clipboard and used in other applications (Expert ...).

In Edit mode, you can delete lines or characters directly in the file comparison, insert them or copy them backwards and forwards between files.
It is also possible to simultaneously and externally process a file opened in the comparison.

7.4.2 Overview of user interface

7.4.2.1 Main menus

The main menus are consistent in Diff. The submenus are dynamically adapted to the associated "SINUMERIK archive comparison", "Folder comparison" or "File comparison" component.

The following main menus and submenus are available in the File comparison:

- File
- Edit
- View

- Comparison
- Tools
- Window
- Help

File

Table 7-18 Diff - File comparison - File main menu

File			
	SINUMERIK archive comparison...		Opens a new SINUMERIK archive comparison.
	Folder comparison...		Opens a new Folder comparison.
	File comparison...		Opens a new File comparison.
	Open	Left-hand side...	Opens a file.
		Right-hand side...	
	Save	Left-hand side	Saves a file.
		Right-hand side	
	Save as...	Left-hand side...	Saves a file with a different name or location.
		Right-hand side...	
	Exit	<Alt+F4>	Exits the Diff program.
	Close	Left-hand side...	Closes the file on the left-hand or right-hand side.
		Right-hand side...	

Edit

Table 7-19 Diff - File comparison - Edit main menu

Edit			
	Cut	<Ctrl+X>	Removes the selected objects and copies them to the clipboard.
	Copy	<Ctrl+C>	Copies the selected objects to the clipboard.
	Paste	<Ctrl+V>	Pastes objects from the clipboard to the selected position.
	Paste before selection	<Ctrl+Alt+V>	Pastes the data contained in memory in front of the selected line.
	Select All	<Ctrl+A>	Selects all lines of the file.

Edit			
	Edit mode	<Ctrl+E>	Lines can be copied from one file to another file only in activated **Edit mode**. (Copy to right / Copy to left). All shortcut menu functions, such as Cut, Paste, Paste before selection and Delete lines, require activation of Edit mode. This is done by clicking the button with the same name. The button has a yellow background when Edit mode is activated.
	Paste new line	<Paste>	Pastes a new line above the selected line.
	Delete line	<Del>	Deletes the selected line.
	Copy to right	<Alt+Right>	Copies the selected line from the left-hand file to the right-hand file.
	Copy to left	<Alt+Left>	Copies the selected line from the right-hand file to the left-hand file.
	Go to next difference	<Ctrl+N>	Displays the differences in direction, forward or backward.
	Go to the previous difference	<Ctrl+B>	

View

Table 7-20 Diff - File comparison - View main menu

View			
	View only left occurrences	<Ctrl+Shift+L>	Displays the parameters available only in the **left-hand** file of the file comparison.
	Different	<Ctrl+Shift+D>	Displays the lines that are **unequal**. The display can be activated or deactivated.
	Equal	<Ctrl+Shift+E>	Displays the lines that are **equal**. The display can be activated or deactivated.
	View only right occurrences	<Ctrl+Shift+R>	Displays those lines available only in the **right-hand** comparison object. The display can be activated or deactivated.
	View differences in context		Groups larger blocks of identical lines in the display and hides most of them to ease finding the differences in the context.
	Toolbars	Standard	Like the search, the standard toolbar can be shown or hidden. The displays are activated as default.
		Search	

Comparison

Table 7-21 Diff - File comparison - Comparison main menu

Comparison				
	Refresh	<Ctrl+D>	The "Refresh comparison" button is always color-highlighted when the displayed comparison result is not current and the refresh of the comparison should be restarted.	
	HTML protocol		HTML protocol	Clicking the HTML protocol button displays in the standard browser (Internet Explorer) the comparison result of the node selected in the data structure. The result can be saved either as HTML or XML protocol.
			HTML protocol data fields as displayed	
			XML protocol	
			XML protocol data fields as displayed	
	Edit configuration...	<Ctrl+M>	Clicking the "Edit configuration..." button opens the "Configuration - file comparison" window. A comparison configuration can be changed or a different comparison configuration selected. More information is provided Section "Configuration - file comparison (Page 189)".	
	Configuration selection		Standard	Switches the comparison configuration directly without dialog.
			HMI configuration (*.ini)	
			NC program (*.mpf; *.spf)	
			BD Data (*.tea)	
			NC data (*.tea; *.ini)	
			NC definitions (*.def)	
			Mask definition (*.com)	
			Mask text (*.com)	
			Help system (*.com)	
			Alarm txt (*.com)	
			UPShield Logbook (*.htm)	
			SciShield Logbook (*.htm)	
			CmcShield Logbook (*.htm)	

Tools

Table 7-22 Diff - File comparison - Tools main menu

Tools			
	Options...	Opens the "Options" dialog with the tabs described below. Options can be activated or deactivated for representation/execution of the Diff components. All **options** are activated as **default**. The Diff coloration and the language setting can be changed user-defined.	
		General	The following settings can be activated or deactivated: • Recompare objects automatically when the comparison configuration changes • Display quick start when Diff starts
		Folder comparison	**Automatic selection of comparison configuration** The following settings can be deactivated or activated: • Select the "SINUMERIK archives" configuration for the SINUMERIK archive comparison • Message for automatic configuration change • Compare NCU folders with different names in the archive as having the same name. (NCU_* == NCU_*) Restart required
		File comparison	**Automatic selection of comparison configuration** The following settings can be activated or deactivated: • Identify file type and select configuration • Message for automatic configuration change **File identification configuration...** Clicking this button opens the "Configuration - file identification" dialog. You will find more information about this in section "Configuring

Tools			
			automatic file identification (Page 195)."
		SINUMERIK archive comparison	**Parameter** The following setting can be activated or deactivated: • Display, copy and export with parameter short name
		Colors	The color setting for the colored line identification of the comparison results can be changed user-defined. This is useful, for example, for a red-green sight impediment. The standard setting can be restored: • Unequal - Red • Equal - Black • Right - Blue • Left - LimeGreen
		Language	Language selection of the Diff interface. Available languages are: • German • English

Tools			
	Data comparison	Data comparison configuration...	Opens the "Configuration - NC data comparison" dialog. Information on this can be found in Section "Configuration (SINUMERIK archive comparison) (Page 127)".
		Configuration data task	Opens the "Configuration - data task" dialog. Information can be found in Section "Data tasks configuration (Page 140)".
	Folder comparison	Folder comparison configuration	Opens the "Configuration - folder comparison" dialog. Information can be found in Section "Configuration (Folder comparison) (Page 169)".
	File comparison	File comparison configuration...	Opens the "Configuration - file comparison" dialog. Information on this can be found in Section "Configuration (file comparison) (Page 189)".
		File identification configuration...	Opens the "Configuration - file identification" dialog. Information can be found in the "Automatic file identification configuration (Page 195)" section.

Tools		
	Export configurations	The following configuration types and the associated files can be selected or deselected in the "Export configurations" dialog: • Archive/folder comparison configurations • File comparison configurations • Data comparison configurations • File identification configurations • Data tasks Only the selected configurations are exported.
	Import configurations	Configurations are imported via the path selection.

Window

Table 7-23 Diff - File comparison - Window main menu

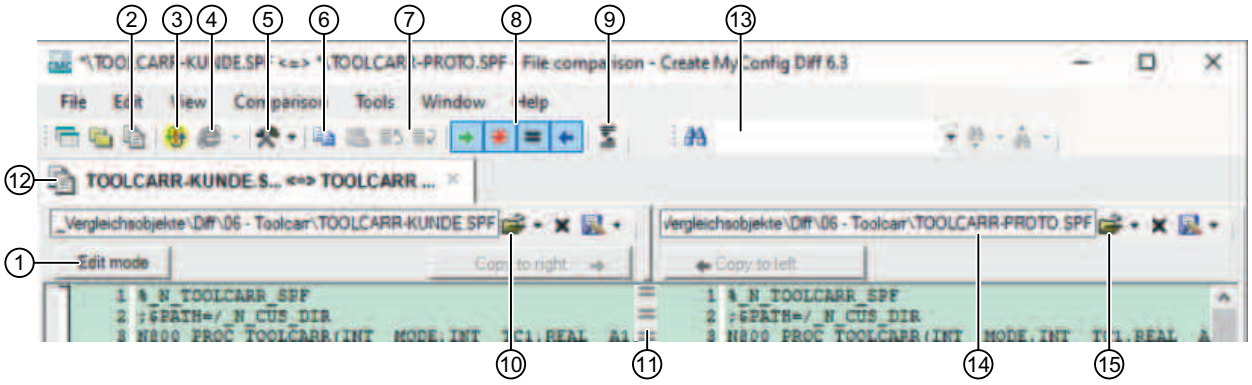
Window		
	New horizontal register group	Displays the selected component in the new horizontal register group.
	New vertical register group	Displays the selected component in the new vertical register group.
	Move to previous register group	Moves the selected component to the previous register group.
	Move to next register group	Moves the selected component to the next register group.
...	...	Displays further opened windows with icon and designation.

Help

Table 7-24 Diff - File comparison - Help main menu

Help			
	Content...	<F1>	Display of the online help for the **Diff** program. The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	CMC script...		Display of the Online Help "CMC script" The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	FAQs...		Display of the Online Help "FAQs" The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	Display quick start...		A SINUMERIK archive comparison, Folder comparison, File comparison or the online help can be opened with a click from the "Quick start" window.
	About...		Contains information on the version and copyright for Create MyConfig Diff.

7.4.2.2 File comparison - toolbar



① Buttons "Edit mode", "Copy to right", "Copy to left"
② New file comparison
Calling a file comparison.
③ Refresh comparison
④ Display/save HTML/XML protocol
⑤ Configure comparison
⑥ Copy
⑦ Go to previous / next difference.
⑧ Group of buttons for displaying and hiding the comparison results.
⑨ View differences in context
⑩ Select file for comparison
Selecting the **left-hand** comparison object
⑪ Column with the comparison result icons, to separate the left-hand and right-hand files.
⑫ Switch between opened comparisons (here, only 1 opened tab)
⑬ Input field Search
⑭ Path of the opened file.
⑮ Select file for comparison
Selecting the **right-hand** comparison object.

Description of buttons/fields of the toolbar

The buttons of the toolbar are described in more detail below.

① Buttons Edit mode / Copy

You can activate and deactivate the Edit Mode using the "Edit mode" button.

Using the "Copy to right" and "Copy to left" buttons, all of the selected lines are transferred from one file into the other.

When the Edit mode is activated, using the shortcut menu, additional edit functions can be used.

You can select, delete, copy and reinsert lines. Further, you can change the line text. You change the display mode using the shortcut menu.

The changes are saved in the file using the floppy disk icon.

③ **Refresh comparison**

The refresh button is always shown in color if the displayed comparison result is not a current result.

Click on the button to update the comparison.

④**Display/save HTML/XML protocol**

The comparison result is shown in the standard browser (Internet Explorer).

You save the result either as HTML or XML protocol.

The comparison result is output completely or, under "...as displayed," the output depends on the settings made under ⑧.

⑤ **Configure comparison**

Using this button, you can change a comparison configuration or select a different comparison configuration.

More information is provided in Section "Configuration (Page 189)".

Note

An automatic selection of the comparison configuration as a function of the file type identification is the default setting, which you can manually change at any time.

You will find more information about this in section "Configuring automatic file identification (Page 195)."

---

⑥ **Copy**

Button to copy selected lines of the left-hand or right-hand file.

More information on this topic is provided in Section "Copying from the file comparison (Page 197)".

⑧ **Group of buttons for displaying and hiding the comparison results**

You can hide or display the associated lines (left only, unequal, equal and right only) of the comparison result using this group of buttons.

⑨ **View differences in context**

The "Differences in the context" button hides larger blocks with identical lines.

If more than 3 identical lines are located before or after a different line, then these (up to 3 lines) are hidden.

⑩ **Select file for comparison**

Here you select the **left** comparison object.

→ Selection of the file to be compared

→ Selection from the files that were last selected

⑫ **Switch between opened comparisons**

In the opened tabs, similar to the Windows task bar, you can toggle between the opened comparisons with a mouse click or by simultaneously pressing "Ctrl" and "Tab".

⑬ **Search input field**

Edit box for the "Search" dialog box to search for text within both files

This group contains:

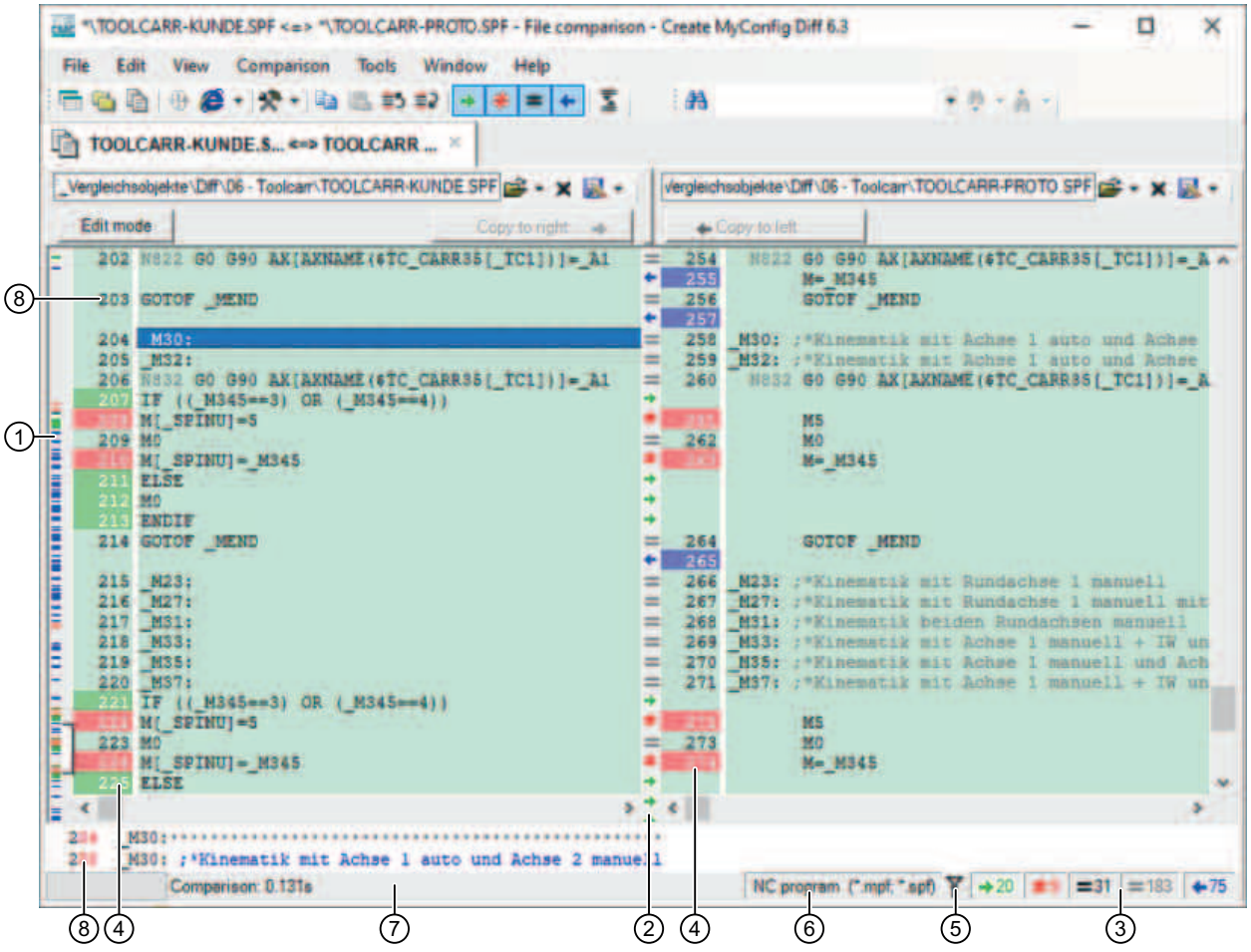
- A button for opening a dialog, in which you can define the search options
- An edit box for searching for character strings in the display fields
- Buttons for searching upward and downward.

 → Search downward

 → Search upward

7.4.2.3 File comparison - overview window

Below is an overview of the functions in the "File comparison" window.



① Overview comparison result
② Column with the comparison result icons, to separate the left-hand and right-hand files.
③ Specified values for lines (status bar bottom right)
④ Comparison result icons and color coding of lines
⑤ Comparison configuration - filter
⑥ Display of the current comparison configuration
⑦ Status bar
⑧ Display of the lines to be compared one on top of the other

Description of the areas/functionalities in the File comparison window

The individual functionalities of the file comparison are described in more detail below.

① Comparison result overview

The comparison result is shown over the complete content of both files in this bar as an overview.

The colored dashes represent lines that are only available on the left-hand side, only on the right-hand side – or are not equal.

The white bars in the right-hand part of the list indicate which section of the comparison result is displayed in the area ④. The bars and therefore the visible section of the window are positioned by clicking on the toolbar.

② Column with comparison result icons

The descriptions for the comparison result icons can be found at ④.

③ Specified values for lines (status bar bottom right)



The values of the status bar provide information on the number and status of the lines in the file comparison:

Color display	Meaning of the lines in the file comparison
Green	Only available in the left-hand file.
Red	Content is different.
Black	Both files are identical.
Gray	Identical in the relevant part of the line.
Blue	Only available in the right-hand file.

④ Comparison result icons and color coding of lines

The lines of the files to be compared are displayed next to one another in this area and are color-coded to make it easier to interpret the comparison result.

- The green arrow and the line number with green background indicate that this line only exists in the left-hand file.
- The not equal sign and the line numbers with red background indicate that the content of the line is not identical.
- The black equals sign indicates that both lines are identical.
- The gray equals sign indicates that the relevant part of the line match.
- The blue arrow and the line number with blue background indicate that this line only exists in the right-hand file.

Note

Non-relevant parts of a line, e.g. spaces or comments, can be defined in the comparison configuration.

More information can be found in Section "Configuration (Page 189)".

Note

Lines or characters shown in gray have been excluded from a comparison by a filter.

⑤ Configure comparison - Filter

A filter is active in the comparison configuration. The comparison configuration is opened with a double-click.

Note

You can exclude character strings that are not relevant for the comparison with inclusion or exclusion filters.

These character strings are shown in gray in the comparison result.

⑥ Display of the current comparison configuration

The comparison configuration that is being used in the current comparison is displayed in this field.

The comparison configuration is opened with a double-click.

⑧ Display of the lines to be compared above one another

The two lines below the comparison window represent the lines located side by side in the comparison window, above one another; this simplifies comparing the lines character by character.

In this display, you can see which type of characters are contained in the lines:

Color display	Meaning of the characters in the lines
Red	Not equal to
Black	Equal to
Green	View only left occurrences
Blue	View only right occurrences

A gray point · represents a character that does not exist in this line. However, there is a character at this position in the other line, which can also be a space.

7.4.3 Configuration

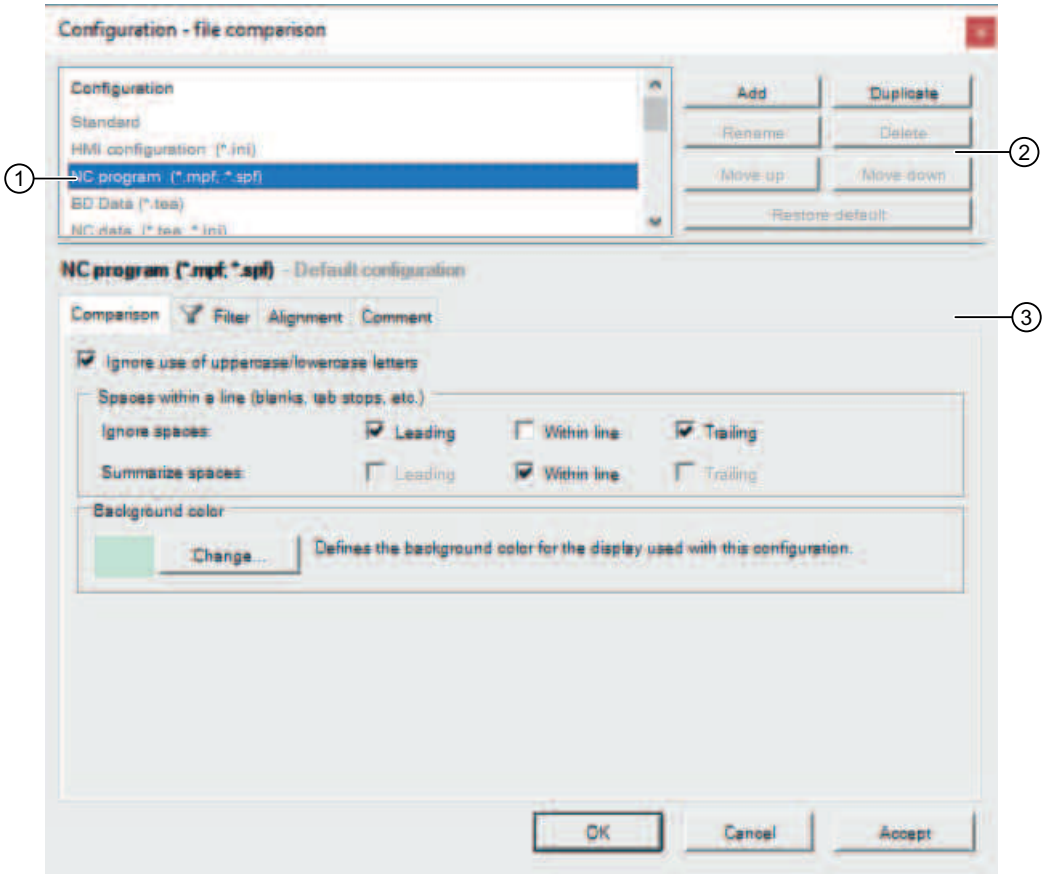
File comparison

You can call the dialog to configure a file comparison using the "Configure comparison" button or via the menu "Comparison" > "Edit configuration...". You can manage the comparison configurations, i.e. create, edit, delete, in the dialog.

All criteria to be considered for a file comparison are defined in a comparison configuration.

You can also export or import configurations, for instance, to make them available for other users.

More information is provided in Section "Exporting and importing configurations (Page 198)".



① The existing comparison configurations are listed in this area of the configuration dialog and can be selected by clicking them. Configurations shown in gray are user-specified configurations; these can be changed, but not deleted - and can be restored again at any time.

② Configurations can be managed using these buttons.

Using "Add", a new configuration is generated, which can also be adapted.

A copy of the selected comparison configuration is created by clicking "Duplicate".

With "Restore default", the changes to the specified comparison configuration are undone.

NOTE:

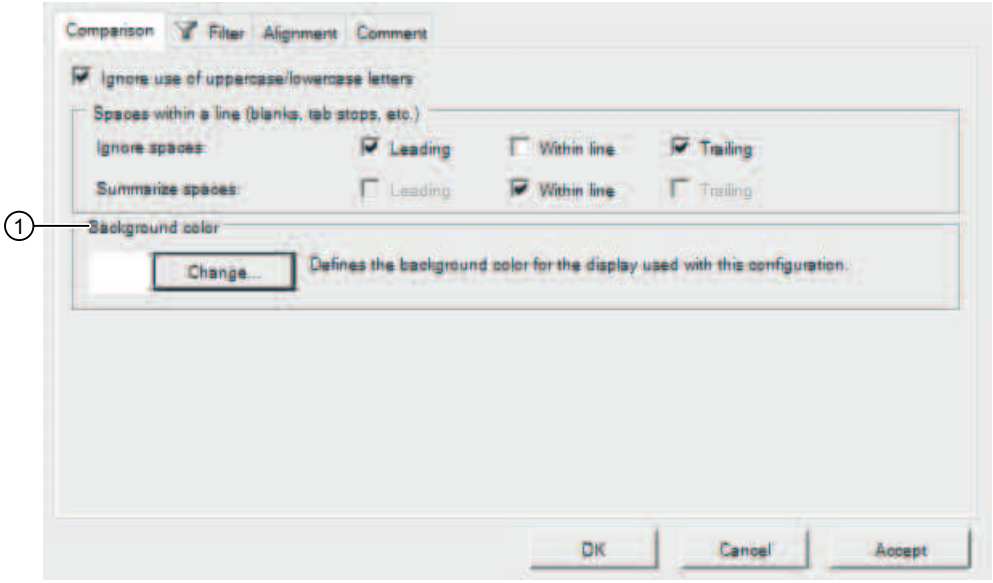
The "Restore default" button is only active if a user-specified configuration has been selected, the contents of which are no longer the same as the default setting. In addition to restoring the default setting, this button can also be used to check whether the contents are identical with those of the default setting.

③ The properties that belong to the selected configuration are stored in this area of the Configuration dialog; these properties include comparison criteria, filter criteria, alignment of the lines and a comment.

Comparison tab

You can define how the file comparison handles upper/lower-case letters and spaces under tab "Comparison".

In addition, you can define the background color for comparisons.



① **Background color**

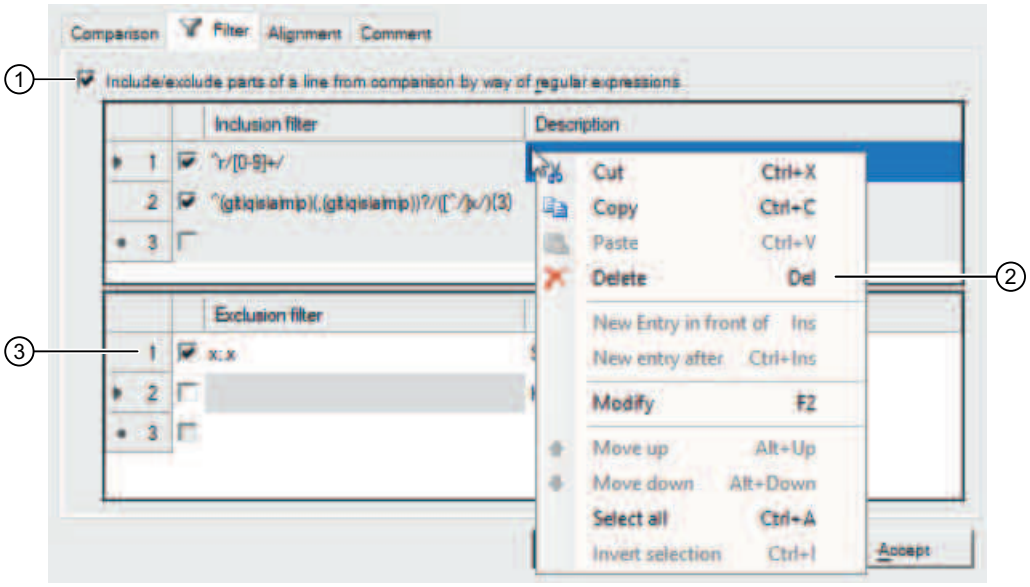
So that users can immediately recognize how to select certain comparison configurations, for the data field, you can define a separate background color for each comparison configuration.

Filter tab

Using filters, non-relevant parts of a line can be included in or excluded from the comparison based on regular expressions.

The characters excluded from the comparison are shown in gray in the comparison result. The content of a file is always displayed completely; however, marked in color.

If only the relevant characters are identical in the line but non-relevant characters are different, the comparison icon will be a gray equals sign.



① The function of the inclusion and exclusion filter is activated using a check mark for "Parts of a line...".

NOTE:

Filter icon is shown in the status bar if this function is activated.

② You can create, edit and delete filter entries using the shortcut menu.

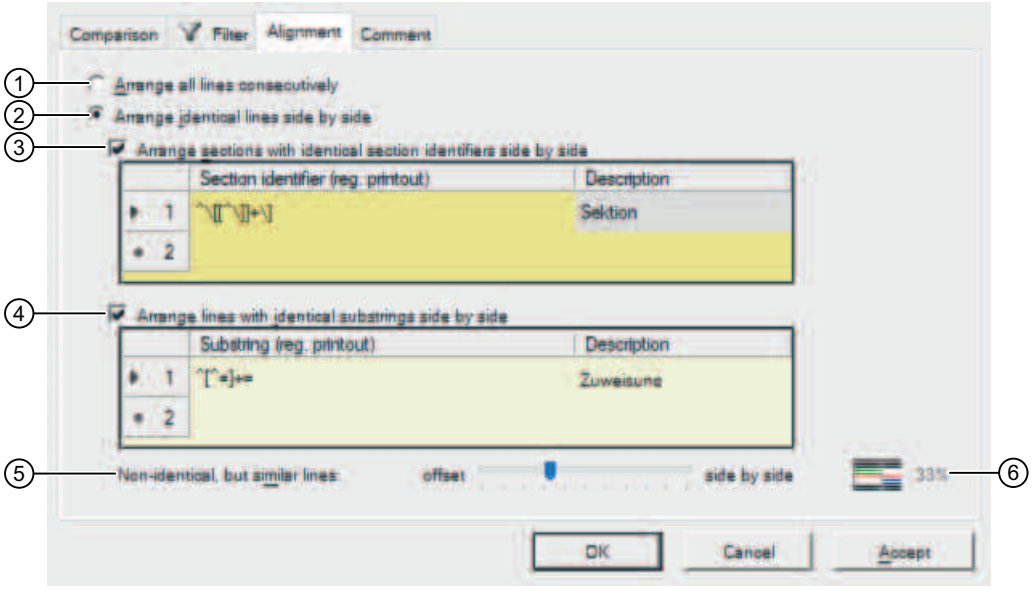
③ Every filter entry can be individually activated or deactivated.

Examples of filters:

;.	The regular expression, which is used for ③, comprises the following characters: A space, an asterisk, a semicolon, a period and an asterisk. **The characters have the following significance**: • Space stands for itself • Asterisk → is a quantifier for any number of preceding characters • Semicolon → stands for itself • Period → stands for any character If this expression is used as exclusion filter, then it excludes all of the semicolons in the file and the following characters in this line that can be of any type. Comments identified using a semicolon are displayed, but are excluded from the comparison. More information is provided in Section "Using regular expressions in Diff (Page 206)".

Alignment tab

In the "Alignment" tab, you define how the individual lines of both files should be aligned with one another after the comparison.



① Using this selection, you can define that all lines of the two files are aligned continuously next to one another (line 1 next to line 1, line 2 next to line 2 ...).

A search is not made across lines regarding identity or similarity.

② As an alternative to ①, this selection that defines that identical lines should be aligned next to one another, is more frequently used in practice.

This selection is refined and/or expanded by specifying ③, ④ and ⑤

③ In the files to be compared, you can define sections that should be taken into consideration and for identical section identifiers, these should be aligned next to one another as priority.

When the option is selected (using the shortcut menu or double-click), a regular expression should be saved in the entry field to describe the section identifier.

NOTE:

Lines found also have a yellow background in the comparison display.

④ Further, you can define that in the files to be compared, lines with identical partial character strings are aligned next to one another as first priority. When the option is selected, a regular expression should be saved in the entry field to define the partial character string.

For example, this function is used to define that in files that have the structure "identifier= value", only identical identifiers are placed next to one another (juxtaposed).

NOTE:

The lines found also have a light yellow background in the comparison display.

NOTICE:

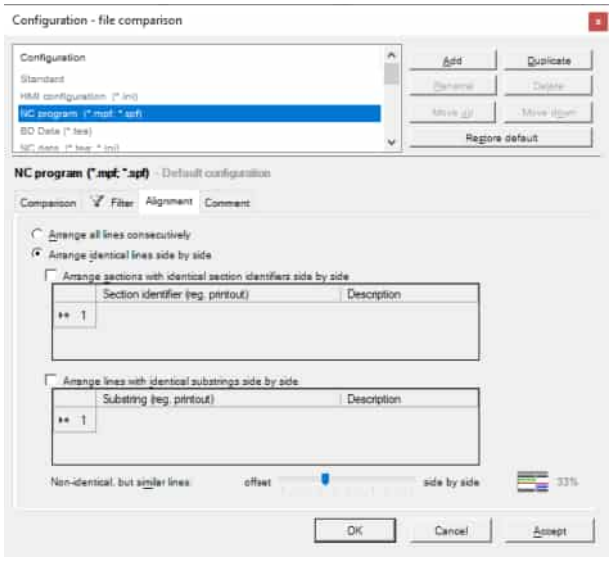
When selection ③ is activated, lines with identical partial character strings are only placed next to one another (juxtaposed) if they are located in the same section.

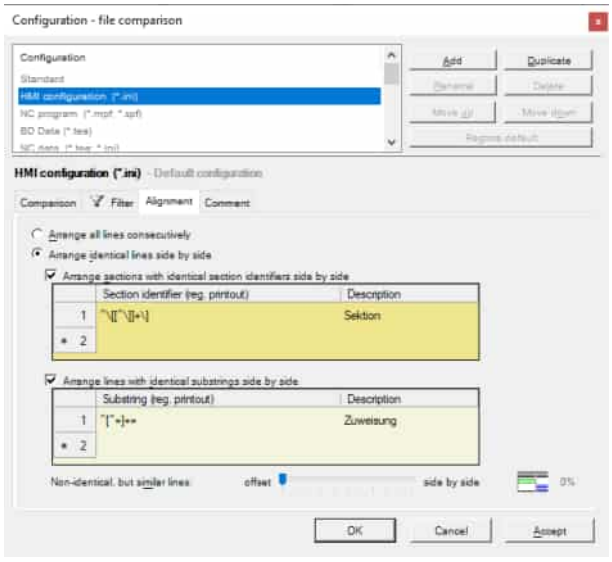
⑤ In addition, for lines for which the definitions ③ and ④ do not apply, this slider sets how the system handles lines that are not identical, but similar when it comes to aligning them. This means it is defined as to whether similar lines, depending on the degree of similarity, are shown offset to one another (green and blue arrow) or next to one another (red unequal icon).

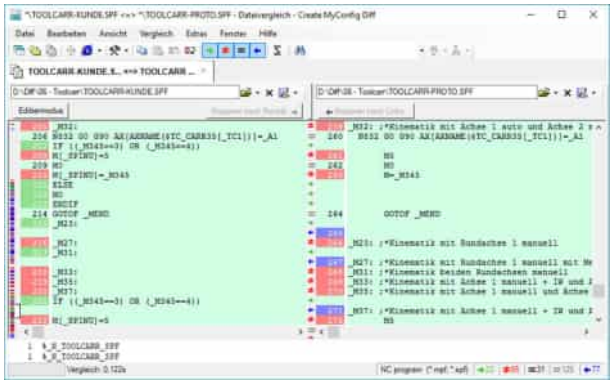
⑥ This diagram symbolically shows how the slider setting functions.

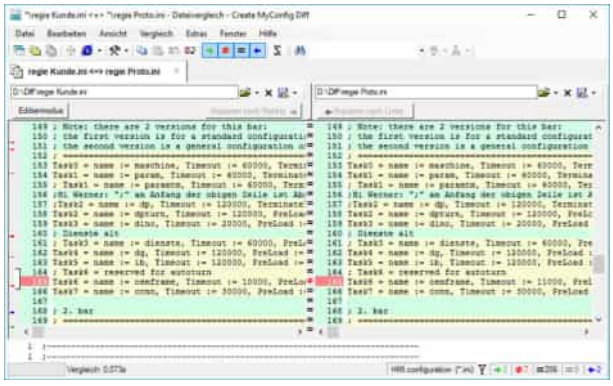
Examples

Two typical comparison configurations for SINUMERIK are subsequently shown and their comparison results as example.









Comment tab

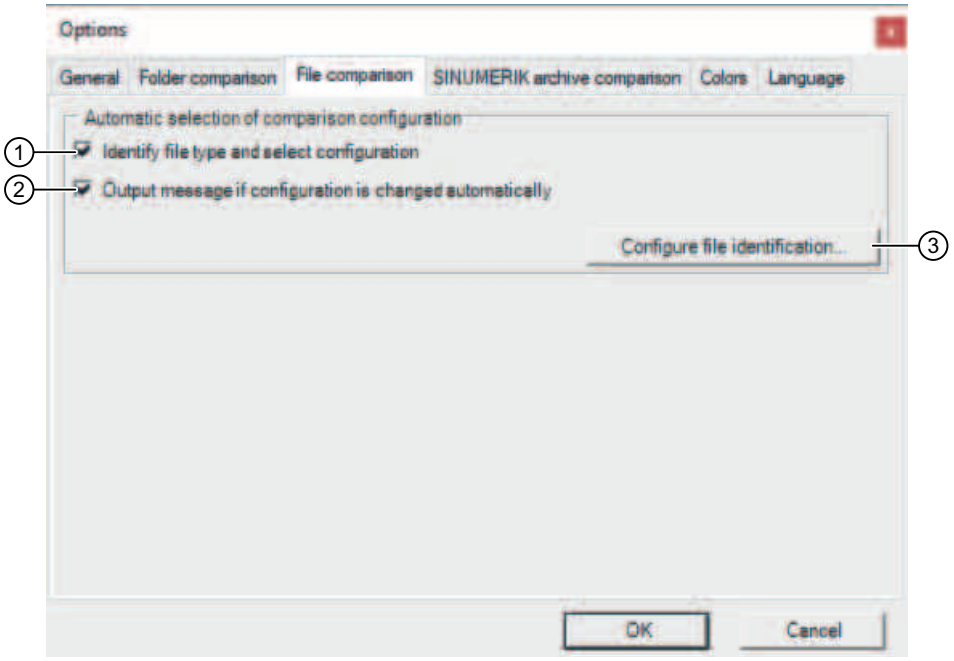
Any comments for the selected comparison configuration can be saved in the "Comment" tab.

7.4.4 Configuration of the automatic file identification

Automatic file identification in Diff

An automatic file identification for file comparison is integrated in Diff. The file identification identifies the file type based on its file name or file content, and selects an appropriate comparison configuration.

You call dialog "Options" to set the program options using menu "Tools" > "Options".



① Deactive/activate configure comparison
② Message for automatic change of the comparison configuration
③ File identification configuration...

Description of the functions in the "File comparison" tab

① Deactivate/activate is active in the Configure comparison

The automatic selection of the comparison configuration is deactivated or activated in the "File comparison" tab.

② Message for automatic change of the comparison configuration

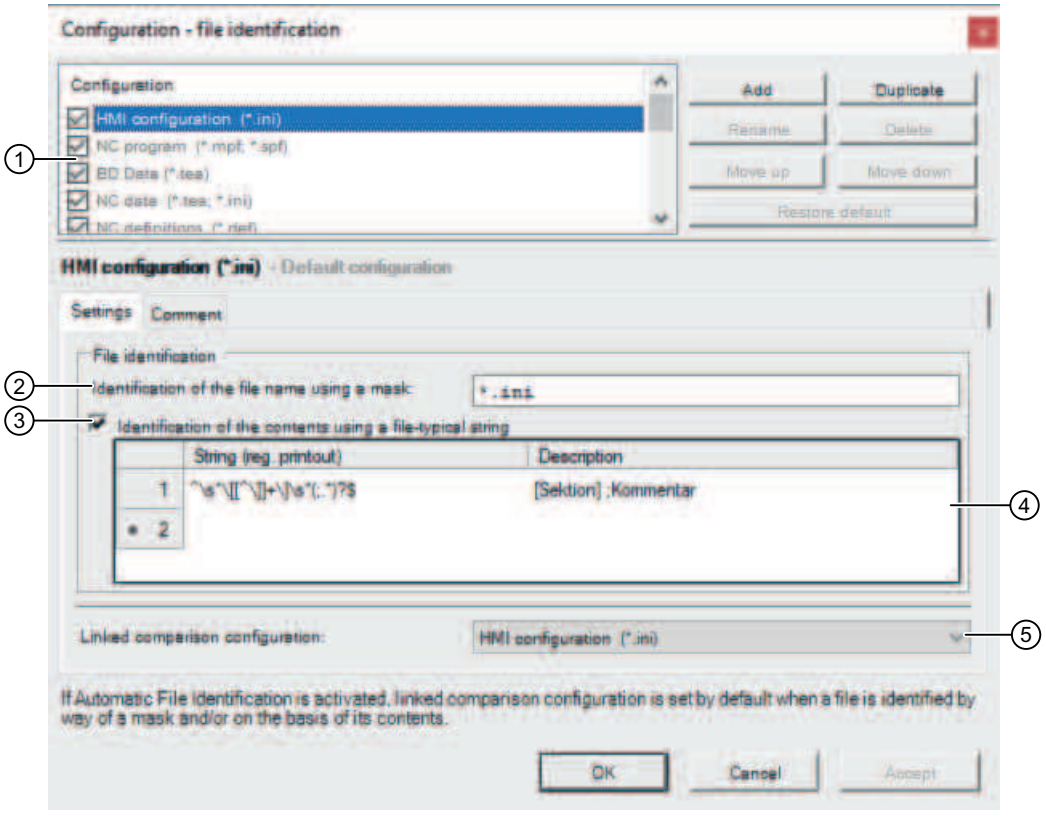
Setting whether a message should be issued when a comparison configuration is changed automatically as a result of the file identification.

③ File identification configuration...

Clicking the "File identification configuration..." button calls the dialog for configuring the automatic file identification.

Configuration - file identification

The dialog for configuring the automatic file identification has the same layout as the dialog for comparison configuration and should be operated similarly.



① List of stored configurations

Tab "Settings":

② Identification of the file name via screen form

③ Activate the "Identification of the content based on a character string typical for the particular file type"

④ Store regular expressions that describe character strings typical for the particular file type

⑤ Linked comparison configuration

Description of the functions in the "Configuration - file identification" dialog / "Settings" tab

① List of stored configurations

When opening a file, Diff checks the stored identification configurations from top to bottom for an appropriate configuration. The 1st appropriate configuration that matches the configuration in tab"Settings" is used. Consequently, the sequence of the stored configurations is significant.

For example: A configuration that operates only with file identification using the file names, should be located after a configuration that operates with file identification using the same file names and also a file content.

You deactivate configurations that are temporarily not to be used by removing the checkmark.

② Identification of the file name via screen form

You can save a screen form to identify the file type based on file names in this field.

③ Identification of the content based on a character string typical for the particular file type

If the file name does not provide an unambiguous identification, an identification is made based on a character string typical for the particular file type.

④ Store regular expressions that describe character strings typical for the particular file type

When activating the content identification via ③, a regular expression that describes a character string typical for the particular file type must be stored in this field. You can specify a regular expression for each line.

When an appropriate regular expression is found, i.e. a defined character string was found in the file, the comparison configuration is used.

⑤ Linked comparison configuration

This list box specifies which comparison configuration should be used when a file described using ②, ③, and④ is identified. The drop down list contains all of the file comparison configurations.

More information can be found in Section "Configuration (Page 189)".

Note

For the user-specified configurations, the "Configuration - file identification" and the associated "Configuration - file comparison" have the same name. It is not mandatory that the names are the same. The configuration names have no significance from a functional perspective.

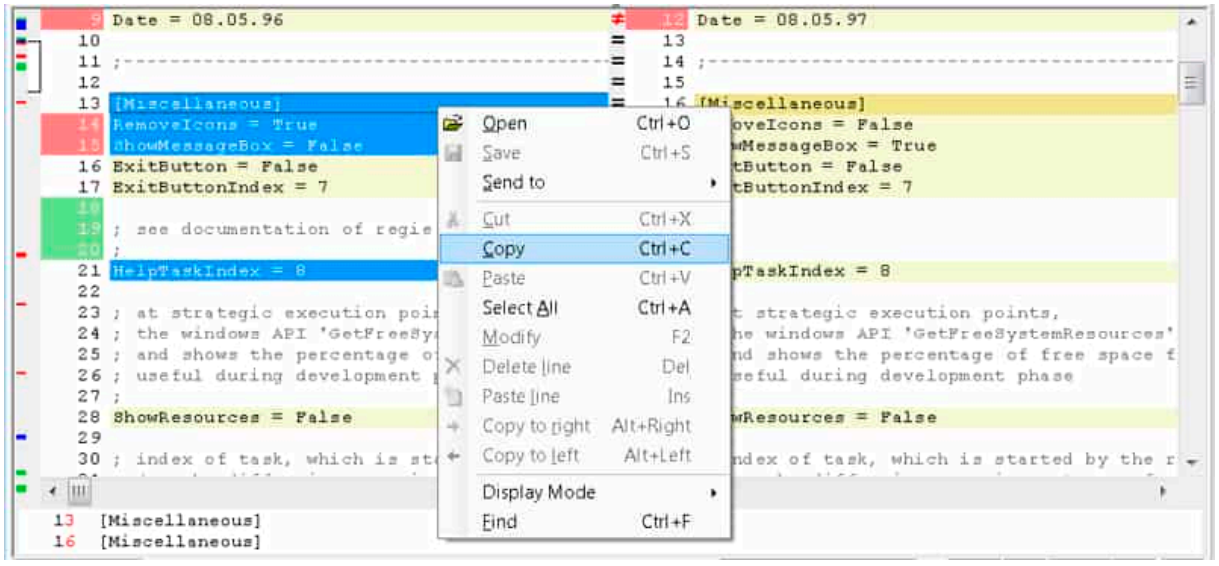
"Comment" tab

Any comments for the selected identification configuration can be saved in the "Comment" tab.

7.4.5 Copying from the file comparison

Contents can be copied line-by-line from the files selected for comparison. You can insert these contents in a component, e.g. in Expert.

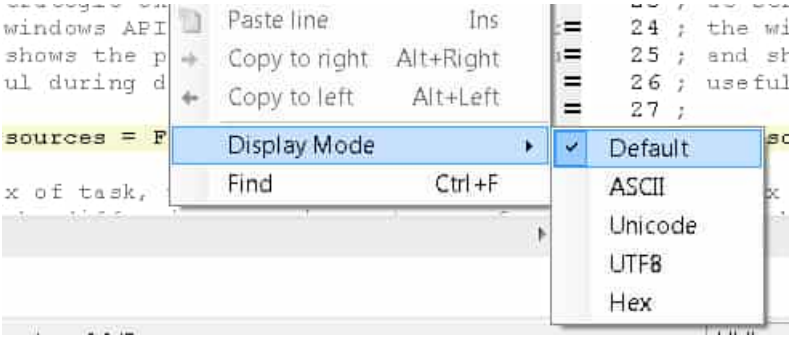
The lines to be copied are selected by left-clicking - keeping the <Ctrl> or <Shift> key pressed at the same time.



Objects are copied using the normal Windows tools (context menu, button, <Ctrl+C>, etc.).

7.4.6 Display modes

The display mode for the file to be compared can be selected using the context menu of the file comparison.



7.5 Exporting and importing configurations

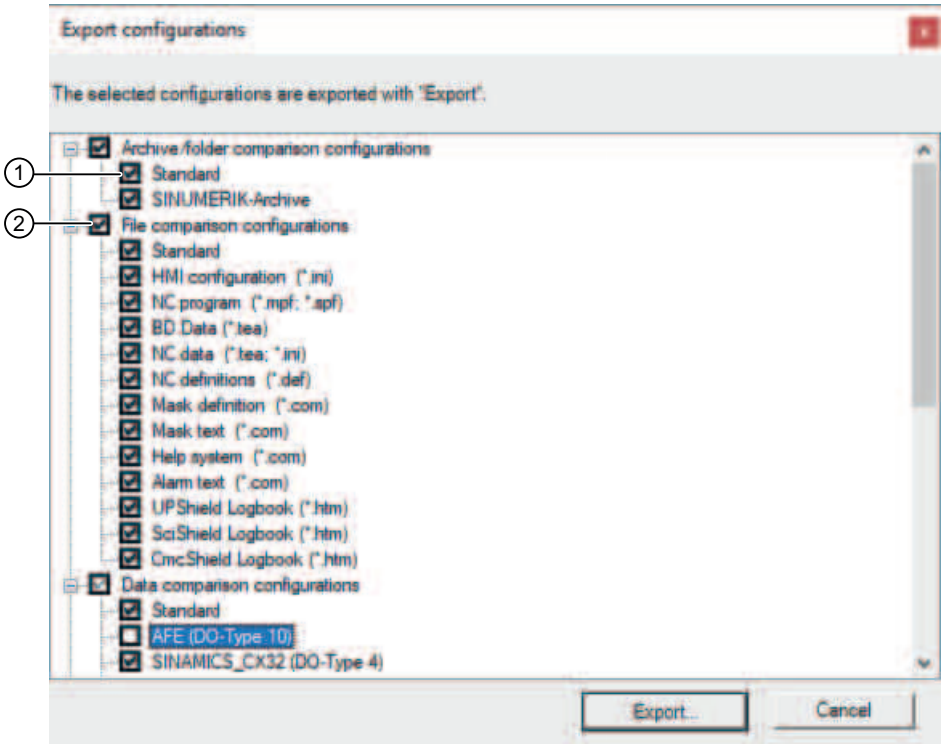
7.5.1 General information

You can export the comparison, file recognition and data order configurations specified in Diff and the ones you created yourself to a file individually or in groups and import them from a file.

This means that you have the possibility of backing up your own configurations outside of Diff and also making these available to other users.

7.5.2 Exporting

You call the dialog to export configurations via the main menu "Tools" > "Export configurations".



① By activating the individual configurations, for each configuration you can define whether this should be exported.

② By activating/deactivating a group node, you define as to whether all group configurations should be accepted.

All configurations that are available in Diff, are displayed and grouped according to the application areas.

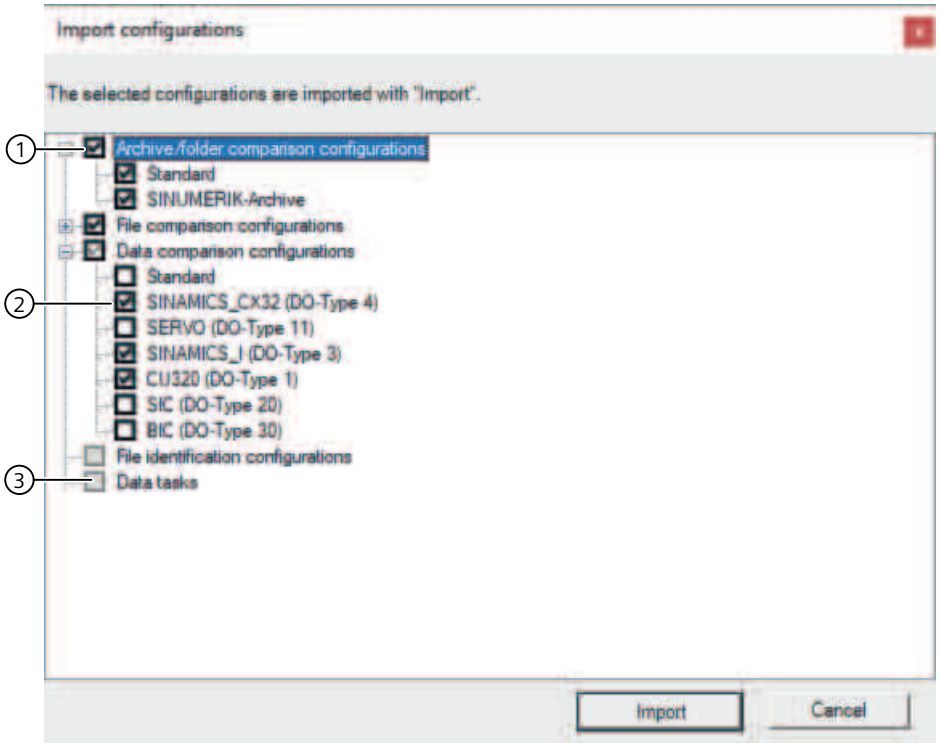
Button "Export..." opens dialog "Save as". The exported configuration file is saved as "*.cfg".

7.5.3 Import

You import configurations via the main menu "Tools" > "Import configurations".

After opening a file with exported configurations, you can select the configurations to be imported in the Import dialog.

The configurations available in the file are displayed and grouped according to application areas.



① Activating or deactivating a group node means that this action is accepted for all configurations of the group.

② By activating the individual configurations, for each configuration you can define whether this should be imported.

③ Group nodes are shown in gray if there is no configuration for the group in the file.

The "Import" button transfers the selected configuration to Diff.

If a configuration is to be transferred to Diff, for which a configuration with the same name already exists in Diff, then you are asked whether the existing configuration should be overwritten.

7.6 Command line call

7.6.1 Diff

Diff has a command line interface so that data can be automatically analyzed and manipulated.

The structure of the command line calls and the principle of operation of the switches and arguments is described in the section.

Call

Write the following commands as well as all deployed switches and arguments to a batch file or in a command line interface in one line.

Call SINUMERIK archive comparison

```
Diff /data <Archive path 1> [< Archive path 2>] [< Archive path n>]
[/importconfigs <Configuration file>] [/config <Comparison
configuration>]
[/compare] [/xml <Log file>] [/html <Log file>][/log <Log file>]
```

Call folder comparison

```
Diff /dir <Directory path 1> [< Directory path 2>]
[/importconfigs <Configuration file>] [/config <Configure
comparison>]
[/compare] [/xml <Log file>] [/html <Log file>][/log Log file>]
```

Call file comparison

```
Diff /file <File path 1> [< File path 2>]
[/importconfigs <Configuration file>] [/config <Comparison
configuration>]
[/compare] [/xml <Log file>] [/html <Log file>][/log <Log file>] [/
encleft <ENCODING>] [/encright <ENCODING>]
```

The significance of the switches and arguments is described in the following "Diff parameters - Significance of the switches" table.

Call, data processing with CMC script

```
Diff /script <Script file> /inarc <Input archive> [/outarc <Result
archive>] [/result <Result file of directory>] [/scriptlog <Script
logbook>] [/log <Log file>]
```

Note

CMC scripts

Observe the following sections for the CMC scripts in Diff:

- CMC scripts in Diff (Page 151)
- Using system variables (Page 34)
- Example of export of archive data - The example is provided in the Operating Manual "Create MyConfig - Diff, Expert, Topo", "CMC script" section, or in the online help with the same name.
- Example of manipulation of archive data - The example is provided in the Operating Manual "Create MyConfig - Diff, Expert, Topo", "CMC script" section, or in the online help with the same name.

Notes

Note the following information for the command line call:

Note

Observe the sequence of the switches

Write the switches in the specified sequence.

When doing this, note the following "Significance of the switches" table.

Note

Path data

- Paths must be specified in straight quotation marks if spaces are included.
  Example:
  `"C:\Path with spaces\Expert_Logfile.log"`
- Paths must never end with '\'.

Note

Saving batch files

Always save batch files with ANSI/ASCII coding.

Note

Umlauts and special characters

Do **not** use any umlauts (special German characters) and special characters when specifying paths.

To allow special characters or umlauts to be correctly processed in specified paths, save the batch files with command line calls with the same character set as used by the region-specific command line interpreter "*cmd.exe".

For example, Codepage 850, also known as DOS Latin 1, applies for the region Western Europe.

The Notepad Editor integrated in Windows does not support saving files with this character set.

Note

Uppercase/lowercase

For the command line switches and arguments, **always** use the specified uppercase/lowercase notation.

You must observe the case sensitivity for the argument of the "/config" switch.

Note

Absolute/relative path

You can specify the absolute path and the relative path.

The relative path with `".\"` or `"./"` refers to the current work folder, where the batch or CMD was called.

If no path is specified, the work folder also applies.

Note

Working without a graphical user interface

Use the "/xml" or "/html" switch to hide the user interface.

Parameter

Significance of the switches

Table 7-25 Diff parameter

Switch	Arguments	Description

`/data`	<Archives>	• Opens the SINUMERIK archive comparison.
		• As arguments, you can specify one or several paths (separated by spaces) for archives or parameter files that are to be compared.
		• A path must be specified in straight quotation marks if spaces are included.
		Example:
		`"C:\path with spaces\NC.ARC"`
		`"C:\path with spaces\NC.dsf"`

`/dir`	<Directories>	• Opens the folder comparison. • As arguments, you can specify one or 2 paths (separated by spaces) for directories or archives that are to be compared. • A path must be specified in straight quotation marks if spaces are included. **Example**: `"C:\Path with spaces\"`
`/file`	<Files>	• Opens the file comparison. • As arguments, you can specify one or 2 paths (separated by spaces) for files that are to be compared. • A path must be specified in straight quotation marks if spaces are included. **Example**: `"C:\path with spaces\MMC.INI"`
`/config`	<Configure comparison>	You can specify the name of a comparison configuration. If the configuration exists, then this is used. If the specified configuration does not exist, or the switch was not used, then the "Standard" configuration is used. **NOTICE** Take the case sensitivity into account. Write the configuration in inverted commas if this contains spaces.

Switch	Arguments	Description
/importconfigs	<Configuration file>	Imports all configurations defined in the configuration file. Existing configurations with the same name will be overwritten. **Note** Select the configuration to be used via the "/config" switch. You create a configuration file in Diff using the "Comparison" > "Edit configuration..." main menu. The configuration file is then exported to a file using the "Tools" > "Export configurations".
/compare	-	The transferred comparison objects are compared.
/xml	<Log file>	Generates a log file of the comparison result in XML format. **Note** If the "/xml" or "/html" switch is specified, Diff works without a graphic user interface.
/html	<Log file>	Generates a log file of the comparison result in HTML format. **Note** If the "/xml" or "/html" switch is specified, Diff works without a graphical user interface.
/encleft	<Coding>	• Only functions in conjunction with the "/file" switch (file comparison). • Specifies the display mode to be used or the coding for the left (first) file. • If the switch is not specified, the coding is selected automatically based on the file content. Possible values are: DEFAULT|ASCII|UNICODE|UTF8|HEX
/encright		Acts just like "/encleft", applies to the right-hand file.

Table 7-26 Switch for the script call

Switch	Arguments	Description
/script	<Script file>	Path of the "*.upscr" script file to be used on the input archive **Note** If the "/script" switch is specified, Diff works without the graphical user interface.
/inarc	<Input archive>	Path of archive "*.ARC"/"*.DSF", to which the script should be applied.

Switch	Arguments	Description
`/outarc`	<Result archive>	Path of the archive that should be created. • Specification of this switch with its parameter is optional. • If the script is manipulated on the input archive, specify this switch. Otherwise processing is not possible.
`/result`	<Result path>	Path of the export data of the script execution. • Specify either a file or a folder. • If several result files are created by the script, specify a path to a folder. • Specification of this switch with its parameter is optional You can find more information under "Using system variables (Page 34)".
`/scriptlog`	<Scripts logbook>	Path of the logbook that was created by Shield during the script execution. • Specify either a file or a folder. When you specify a folder, the logbook is saved in the folder with the file name "<Scriptname>_<Resultarchivename without extension>_<YYYYMMDDhhmmss>.htm" • Specification of this switch with its parameter is optional
`/log`	<Log file>	Path of the log file for the outputs from Diff : • While executing the command line, outputs from Diff are written to the log file in the XML format. • Specification of this switch with its parameter is optional

Note

Path data

You can specify paths either in absolute or relative terms.

Relative path data refers to the location of the batch script that was started.

Note

CMC scripts

More information is provided in Section "CMC scripts in Diff (Page 151)".

Return values

Table 7-27 Return values Diff

Value	Description
0	The comparison returns no differences. The comparison is only executed with switch "/compare".
1	The comparison returns differences.
2	Error (e.g. comparison object not available)
3	Command line call error.

Example

Precondition:

- Create MyConfig is installed in the "C:\Program Files\Siemens\Automation \SINUMERIK\Create MyConfig 6.3" directory.
- Archives "Archive1.dsf" and "Archive2.dsf" are available in path "C:\Temp\".
- The "myConfig" comparison configuration is available in Diff.

```
"C:\Program Files\Siemens\Automation\SINUMERIK\Create MyConfig
6.3\Diff.exe" /data "C:\Temp\Archive1.dsf" "C:\Temp\Archive2.dsf"
/compare /config "myConfig"
```

7.7 Using regular expressions in Diff

7.7.1 General information on regular expressions

Regular expressions represent a general notation to describe text patterns. They are known as a type of programming language to test and manipulate texts.
A regular expression is a character string with syntactic rules which defines significant text patterns or character strings in a general form. Regular expressions are used in Diff to identify a significant character string and to filter it (e.g. all block numbers or all comments in a cycle).

A regular expression comprises 2 types of characters:

- Literal characters
- Meta characters

Literal characters

Literal characters are characters that only represent themselves in a regular expression and have no special significance.

A search with a regular expression that only contains literals, therefore corresponds to a conventional character search.

Meta characters

Meta characters are characters that have a special function within regular expressions.

Some meta characters follow:

$ ^ { } [ ] ( ) | . * + ? \ -.

Similar to natural languages, literals can be considered as being the vocabulary, with meta characters being the grammar of regular expressions.

Escape sequence

If a meta character is required as literal character to formulate a regular expression, i.e. it must only stand for itself, then the character must be masked with what is called an escape sequence. This is achieved by prefixing a backslash "\".

Example

An open square bracket "[" identifies a regular expression.

Insert a backslash "\" in front of the square bracket.

Because the single square bracket is only a meta character, i.e. has a special function, see character classes.

In this case, "\[" is written in the regular expression.

7.7.2 Meta characters in the regular expressions

The most important meta characters are shown in the following subsections which are contained in the dialect of the regular expressions of Create MyConfig.

Character classes

In a regular expression, a character class defines *one* character, which may be *one* of the characters contained in the character class.

Meta characters	Description
.	Corresponds to any character with the exception of \n (line break).
[ ]	A character set available for selection can be defined using square brackets. The expression in the square brackets precisely stands for one character from this selection.

Meta characters	Description
`[acdb]`	Corresponds to *one single* character which is contained in the specified character string. This means one a, one b, one c or one d. The sequence of the characters within the character class is of no significance.
`[^acdb]`	Corresponds to any, individual character which is *not* contained in the specified character string. If a `^` is not located at the beginning of the definition, then it should be considered as literal. For "character negation" the `^` can only be located *within the square brackets* and here, in turn, may only be used at the *beginning*! Outside the square bracket, `^` has a different meaning, see below: "Constructs".
`[0-9a-fAF]`	Adjacent characters can be specified as a range by using a hyphen. If the hyphen is written at the beginning or at the end of the character class or with backslash, i.e. `\-`, then it stands for itself and is not valid as specifying a range.
`\w`	A character which is a letter, a number or an underscore.
`\W`	A character which is neither a letter, number nor underscore.
`\s`	Corresponds to any space character, i.e. space character, tabulator, line feed spectrum.
`\S`	Corresponds to any non-space characters.
`\d`	Corresponds to any decimal number, corresponds to `[0-9]`.
`\D`	Corresponds to any non number, corresponds to `[^0-9]`.
`\t`	Corresponds to a tab stop character.
`\n`	Corresponds to a line break.
`\x20`	Corresponds to an ASCII character in the hexadecimal notation (precisely two positions).
`\u0020`	Corresponds to a unicode character in the hexadecimal notation (precisely four positions).

Constructs

Meta characters	Description
`|`	Several regular expressions can be combined to form a single one using alternation (logic OR). Example: `cat|dog|tiger`. (cat or dog or tiger) The match to the far left has priority.
`( )`	The grouping is used to combine parts of regular expressions and to limit the range of validity of an alternation. Example: `_(abc|xyz)_`
`^`	The meta character for the beginning of the line does not describe a character, but instead a position in a line. Example: `^[aA]` requires a lower-case `a` or an upper-case `A` at the beginning of the line. The `^` character in square brackets has a different meaning, it is used as negation of the character string.
`$`	The meta character for the end of the line does not describe a character, but instead a position in a line. Example: `[aA]$` requires a lower-case `a` or an upper-case `A` at the end of the line.

Quantifier

Using quantifiers, you can specify how many times a specific component of a regular expression must occur in order for a found character string to be considered a match. A quantification expression is valid for the character or character set which is located directly in front of it.

Meta characters	Description
`?`	Requires a null or a match. Example: `_? or (abc|xyz)?`.
`+`	Requires one or any number of matches. Example: `a+ or [0-9]+`.
`*`	Requires null or any number of matches. Example: `\w* or (abc)*`.
`{n}`	Requires precisely n matches. Example: `(abc){2,}`.
`{n,}`	Requires a minimum of n matches, but there can also be more. Example: `(abc){2,}`.
`{n,m}`	Requires at least n matches, however a maximum of m matches. Example: `\w {2.17}`.

7.7.3 Examples of regular expressions

Regular expression	Character	Description
`^\[[^\]]+\]`		Describes a section, e.g. *[Version]* At the beginning of the line there should be an "open square bracket" `[` and then there should be at least one character other than a "close square bracket" `]` followed by a "close square bracket" `]`. The remainder up to the end of the line is not specified.
①②③④⑤...		Numbers indicate the position of the character in the regular expression.
The regular expression is described below from the left to the right:		
^`\[[^\]]+\]`	①^	Here, the beginning of the line, i.e. the described expression should be located at the beginning of the line.
^****`[[^\]]+\]`	②\	Escape character; the "open square bracket" `[` should stand for itself as literal and not as meta character of a character set.
^**[**`[^\]]+\]`	③[	Character stands for itself, i.e. at the beginning of the line there should be an "open square bracket" `[`.
^\[**[**`^\]]+\]`	④[	Starting definition of a character set (character is a meta character).
^\[[**^**`\]]+\]`	⑤^	Here, a negation character, i.e. the characters specified in the character set, should not occur. The character negation is only possible within square brackets, therefore, brackets are used for characters ④ and ⑧.
^\[[^****`]]+\]`	⑥\	Escape character; the "close square bracket" `]` should stand for itself as literal and not as meta character of a character set.
^\[[^**]**`]+\]`	⑦]	This character stands for itself.
^\[[^\]**]**`+\]`	⑧]	End definition of a character set

Regular expression	Character	Description
^\[[^\]]**+**\]	⑨+	Requires one match or as many matches as specified, i.e. there must be at least one character, which is not a "]".
^\[[^\]]+****]	⑩\	Escape character; the "close square bracket"] should stand for itself as literal and not as meta character of a character set.
^\[[^\]]+**]**	⑪]	This character stands for itself.

Regular expression	Character	Description
^[^=]+=		Describes a value assignment (identifier = value) e.g. *Version = 1.0.3* At the beginning of the line there should be at least one character, which is *not* an equal sign "=". This expression (^[^=]+) stands for the "identifier" in the value assignment. This should then be followed by an equal sign "=" and the remainder of the line, the value, is not specified.
①②③④⑤...		Numbers indicate the position of the character in the regular expression.
The regular expression is described below from the left to the right:		
^[^=]+=	①^	Here, the beginning of the line, i.e. the described expression should be located at the beginning of the line.
^**[**^=]+=	②[	Start of the definition of a character set (character is a meta character).
^[**^**=]+=	③^	Here, a negation character, i.e. the characters specified in the character set should not occur.
^[^**=**]+=	④=	Equal sign (literal) stands for itself.
^[^=**]**+=	⑤]	End of the definition of a character set.
^[^=]**+**=	⑥+	Requires one match or as many matches as specified, i.e. there must be at least one character, which is not an equal sign.
^[^=]+**=**	⑦=	Equal sign (literal) stands for itself.

Regular expression	Character	Description
^CHANDATA\(\d+\)		Describes a CHANDATA operation in an NC-INI file, e.g. *CHANDATA(1)* CHANDATA should be at the beginning of the line, followed by a "open round bracket", followed by at least one digit, followed by a "close round bracket".
①②③④⑤...		Numbers indicate the position of the character in the regular expression.
The regular expression is described below from the left to the right:		
^CHANDATA\(\d+\)	①^	Here, the beginning of the line, i.e. the described expression should be located at the beginning of the line.
^**CHANDATA**\(\d+\)	②CHANDATA	This literal character stands for itself, i.e. a search is made for a CHANDATA.
^CHANDATA****(\d+\)	③\	Escape character; the "open round bracket" should stand for itself as literal and not as meta character of a group.
^CHANDATA**(**\d+\)	④(	This character stands for itself.

Regular expression	Character	Description
^CHANDATA\(**\d**+\)	⑤\d	Corresponds to any decimal number.
^CHANDATA\(\d**+**\)	⑥+	Requires one match or as many matches as specified, i.e. there must be at least one decimal number.
^CHANDATA\(\d+****)	⑦\	Escape character; the "close round bracket" should stand for itself as literal and not as meta character of a group.
^CHANDATA\(\d+**)**	⑧)	This character stands for itself.

Create MyConfig - Expert

8

8.1 Introduction

Note

Documentation structure: Operating Manual / Online Help

The contents of Section **Expert** are identical in the Operating Manual and the online help.

Expert is the engineering software for creating a CMC project.

An executable package for producing or upgrading of machines is generated from the CMC project with the "Deploy project" function.

The package contains a configurable sequence of production or upgrading steps and the associated data.

In Expert, you specify the configuration of the package, the dialog for operator interactions and the individual upgrade steps in the form of a hierarchical step tree. You specify the required data or data to be manipulated and files.

Project - creating and configuring

You can configure the majority of the settings easily in the "Project explorer (Page 234)"

- Directly in the dialog "Configuration (Page 236)"
- Under "Dialogs" and "Steps"
- Via a context-sensitive shortcut menu

A "check run (Page 229)" informs you if there are any errors in the configuration.

Script Editor

Expert provides the integrated "Script Editor" which effectively supports you when creating scripts.

Executable package

The "Deploy project" function creates a package that can be executed under Linux or Windows depending on the configuration.

Documentation

An integrated online help (F1), which is identical to this documentation, supports you in the correct configuration.

Note

Qualified personnel

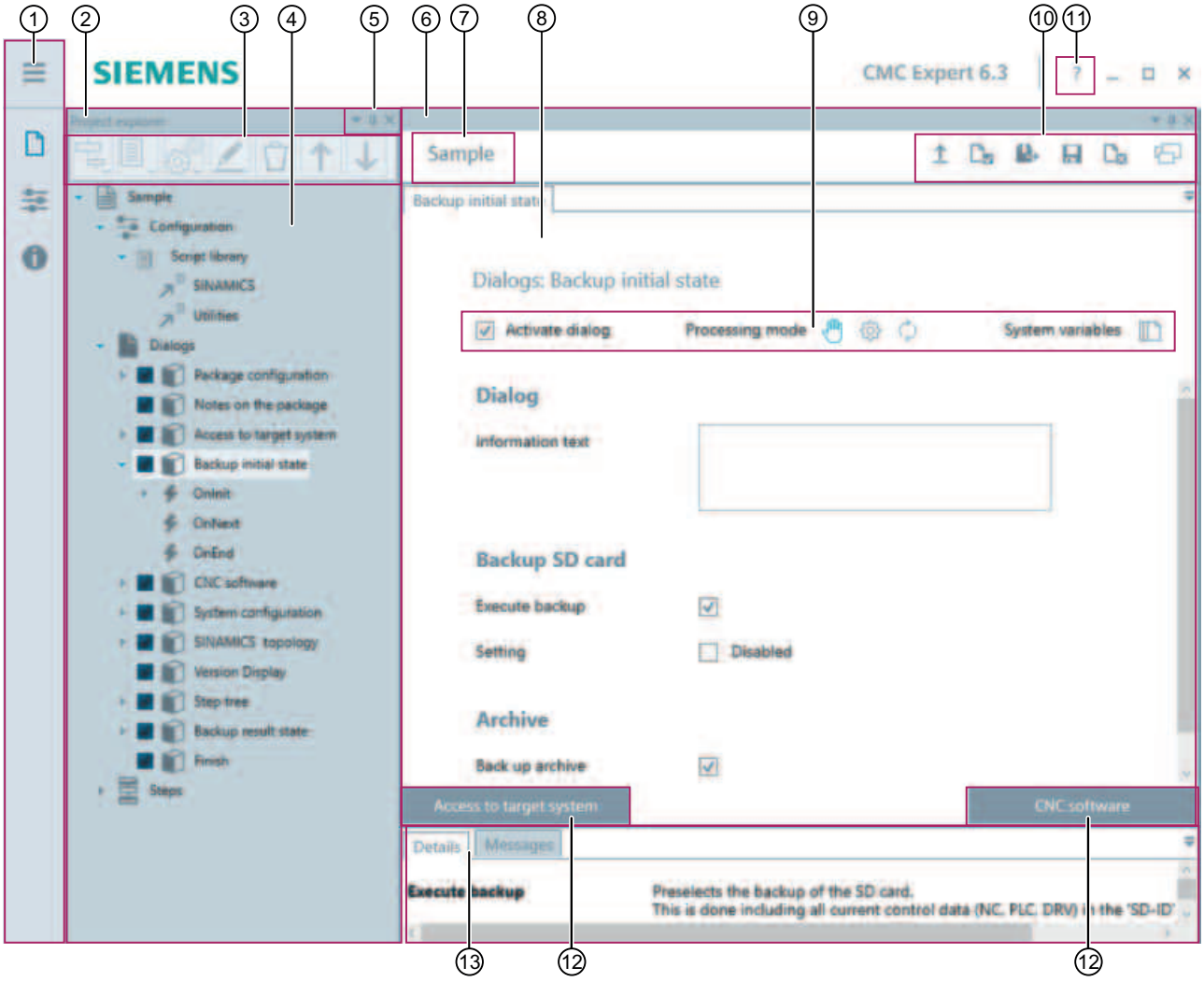
The configuration can only be performed by qualified personnel.

8.1.1 Overview of the Expert user interface

Expert - user interface

The Expert user interface comprises the following areas:

- Left-hand side
  Drop-down menu "Content area":
  - Project
  - Settings
  - Info - Version number
- Center
  "**Project explorer" viewlet**
  - Toolbar with functions for dialogs and steps
  - Opened project with project tree
- Right-hand side
  Home area viewlet:
  - Project name (top left)
  - Buttons for the project (top right)
  - Tab with "Properties" display for configuration, dialog, step, script or action
  - Define dialog properties/call system variables list for the selected dialog
  - Buttons for the previous or following dialog properties
- Bottom right-hand side
  Display viewlet "Messages" or "Details"



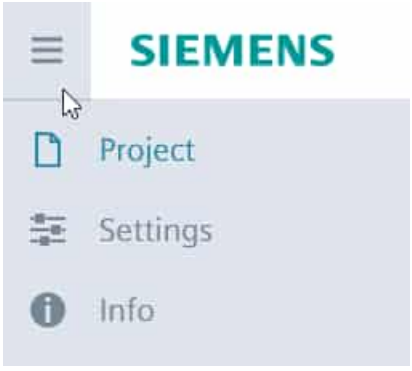
① Drop-down menu "Content area"
② "Project explorer" viewlet
③ Toolbar for dialog and step processing
④ Open project - project tree
⑤ Viewlet - setting the window position
⑥ "Home area" viewlet
⑦ Display of project name
⑧ Display of "Properties" for dialog
⑨ Define basic dialog properties/call system variables for dialog
⑩ Buttons for the project
⑪ Call online help
⑫ Buttons for
Left: previous dialog
Right: next dialog
⑬ "Messages" viewlet / "Details" viewlet (representation of tab)

Figure 8-1 Expert user interface

① Content area

Drop-down menu with the following information:

Table 8-1 Expert - content area

 Content area menu	**Content area** Left-click to open the menu.
	Project Display of the current project
	Settings Language selection Available languages are: • German • English Switch between the interface languages German and English. The changed language is activated when restarting Expert. The online help is displayed according to the interface language.
	Info Version information of the CMC Expert software

②Project explorer viewlet

By default, the "Project explorer" viewlet is displayed at this position in the Expert user interface.

In accordance with your requirements, you can remove the viewlet at this position and dock it to a different position in the Expert user interface. Or you pull the viewlet out of the Expert user interface as a separate window.

③ Toolbar in the Project explorer

You use the functions of the toolbar in dialogs and steps. The usable functions of the toolbar are activated depending on where you are in the project tree. Buttons that cannot be used are grayed out.



Figure 8-2 Toolbar in the Project explorer

Use of the toolbar functions

Dialogs:

- Events
- Scripts

Steps:

- Steps/substeps
- Scripts
- Actions

Table 8-2 Toolbar in the Project explorer

Functions of dialogs	Functions of steps		Description
-			**Add step** Use this button to insert a new step or substep. Open the selection by left-clicking this button.
			**Insert new step** Inserts a new step after the currently selected step.
			**Insert new substep**
			**Add script** (applies to dialogs/steps) • In dialogs below events • In steps below steps or substeps
			**Creating a new script** Click on the element under which you want to create a new script, e.g.: • Dialog > Event (OnInit, OnNext, OnEnd) • Steps > Step or substep The button is activated, click "Create new script".
			**Link an existing script** Click on the element under which you want to link a script, e.g.: • Dialog > Event (OnInit, OnNext, OnEnd) • Steps > Step or substep After clicking on the button, open there to link the required script "*.upscr".
			**Linking an existing script library**

Functions of dialogs	Functions of steps		Description
-			**Add action**
			Creates and links a new action of the type "Copy"
			Creates and links a new action of the type "Change INI"
			Creates and links a new action of the type "Create with pre-processor"
			Creates and links a new action of the type 'Delete'.
			Creates and links a new action of the type 'Execute'.
			Link an existing action
			**Rename** (in dialogs/steps) You can rename the following elements in the project tree: • Scripts • Steps/substeps • Actions Select the appropriate element. Then click on the button to rename the element.
			**Remove from project tree** You can remove the following elements from the project tree: • Scripts (dialogs/steps) • Steps/substeps • Actions Select the appropriate element. Click on the button and confirm with "OK" in the following dialog to remove the element from the project tree. **Note** A script is only removed from the project tree. The script is retained in the source folder.
			**Move up** Move the selected element upwards within a layer
			**Move down** Move the selected element downwards within a layer

④ Opened project - project tree

Project tree structure:

- Project name
- Configuration (Page 236): configuration of the parameters for the package forwarding and the data areas used
- Dialogs (Page 239): the corresponding dialogs are displayed depending on the configuration
- Steps (Page 326): creation of steps, actions and scripts

⑤ Viewlet - setting the window position

Expert contains the following viewlets:

- Project explorer
- Home area
- Details
- Messages

According to the requirements, each viewlet can be moved to another location in the expert user interface and redocked via the positioning tiles. The "Project explorer" and "Messages" viewlets can also be taken from the expert user interface and displayed as a separate window.

	Buttons for viewlets	
	Window position	The following window positions are available: • Cancel anchoring The Viewlet is removed from the expert user interface and displayed as a separate window. • Dock If you select this option, the window is reset to the default position. • Dock as a document in tab format The "Project explorer" and "Messages" viewlets are displayed in the Home area as a tab. To place a tab at a different window position, drag the tab to the desired new position via the positioning tiles by clicking and holding the mouse button. • Automatically in the background • Close
	Automatically in the background	-
	Close	The viewlet is closed and can therefore no longer be accessed via the expert interface. The viewlet can be viewed again after the project has been closed and reopened.

⑥ Home area viewlet

The Home area viewlet contains:

- ⑦ Title bar project name
- ⑧ Display of the properties of the item selected in the "Project explorer", e.g. configuration, dialog or step
- ⑨ Basic dialog settings
- ⑩ Buttons for project execution
- ⑪ Call online help
- ⑫ Buttons for switching over to the previous or subsequent dialog

⑦ Display of project name

The project name is displayed in the title bar. A further display of the project name can be found in the Project explorer.

⑧ Display of properties

Display of the dialog name with display of the associated properties.

⑨ Basic dialog settings

Table 8-3 Basic dialog settings toolbar

☑	Activate dialog	With this check box, you specify whether or not the dialog is used for package execution. Package execution is enabled for this dialog and the following properties are effective only if you have selected the check box.
	**Processing mode**	By default, the Edit mode is set to "Manual". The activated Edit mode is indicated by the blue color. The modes that are not in use are grayed out.

		Manual	The dialog page is completely displayed. The operator can make changes and must trigger dialog page advancing or execution.
		Automatic	The dialog page is completely displayed and executed automatically. The operator cannot make any changes.
		Progress	Processing corresponds to the "Automatic" mode without the configuration information being displayed. A progress indicator is displayed in a small display window.
		System variables	Left-click on the icon for system variables to open the window in which the system variables are listed for the dialog. More information on the system variables can be found in the "Reference list system variables (Page 380)" section.

⑩ Project processing

Table 8-4 Project processing toolbar

			Buttons for the project
		Export configuration to file	This selection allows active configurations to be exported as a file. Opens the "Configuration selection" dialog. Select the configuration areas for the system variables to be exported: • Package • Dialogs • Steps The selected areas are backed up via the "Save" button. Navigate to the desired path/folder and save the exported configuration file with file extension "*.upcfg".
		Review project	With this function, you start the check run. The messages for the check run can be found in the "Messages" viewlet. You can find more information on reviewing the project in the "Check run (Page 229)" section.
		Deploy project	The project is forwarded with this function. If you have not started a check run beforehand, it will be performed automatically before the project is passed on. The messages for the check run are displayed in the "Messages" viewlet.
		Save project	Changes in the project are saved.
		Close project	The project is closed. If you have not saved any existing changes beforehand, a confirmation prompt is displayed to confirm whether the changes are to be saved. The changes can be saved or discarded.
		Show and hide viewlets	Without function

⑪ Call online help

The help is displayed as HTML5 online help. In terms of content, the online help is identical to the operating manual.

⑫ Buttons to navigate to the previous/next dialog

Buttons for

- Left: previous dialog
- Right: next dialog

⑬ Details viewlet/Messages viewlet

Messages tab

After you have executed "Review project" or "Deploy project" in the Home area, information is displayed in the "Messages" viewlet.

Details tab

Click on a property in the Home area to display the detailed description in the Details tab.

8.1.2 Easy to operate Expert user interface

In Expert you adapt the dockable viewlets to your ambient conditions and the work situation.

This is advantageous for larger projects to improve the overview and readability of a project.

By default, the "Project explorer" viewlet is displayed in the center. You can, for example, move the "Project explorer" viewlet to the right and the Home area viewlet to the center.

The content area is always permanently anchored on the left-hand side.

You can display the "Messages" viewlet as a tab and thus switch between the properties displayed in the Home area and the messages as required.

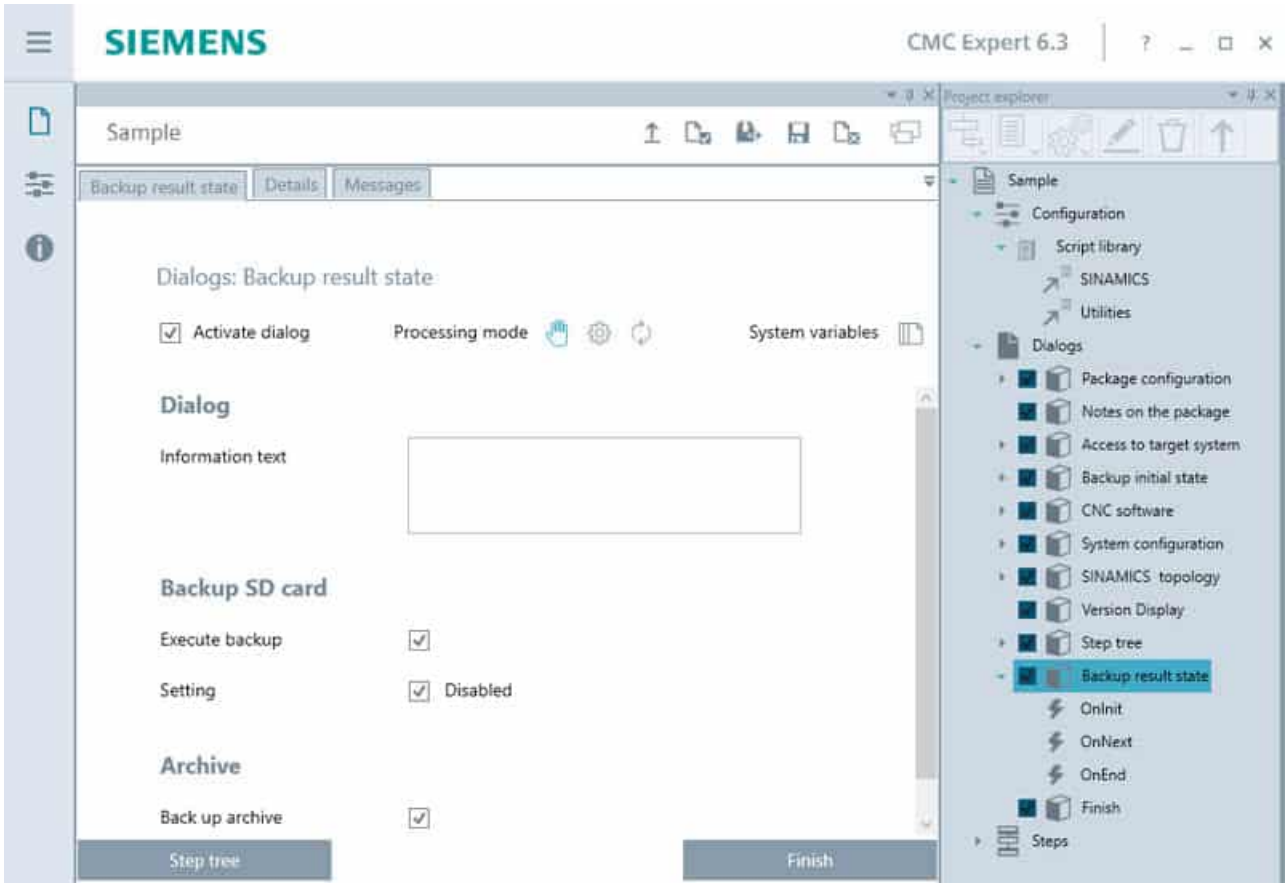


Figure 8-3 Example, changing the expert user interface: "Project explorer" and "Home area" - Swap side view - Messages, display as tab

Dockable viewlets are:

- Project explorer
- Home area
- Messages
  - Display of the Check run (Page 229)
- Details
  - Descriptions of properties from the Home area viewlet

Procedure when resetting dockable viewlets

The following examples show you how you can adapt the viewlet to address your specific requirements.

1. In the default settings of Expert, the content area on the left forms a fixed frame. The "Project explorer" viewlet with the project tree is placed in the center and the Home area viewlet with the display of the properties is located on the right-hand side of the Expert user interface. The "Details" and "Messages" viewlets are located below the Home area viewlet.

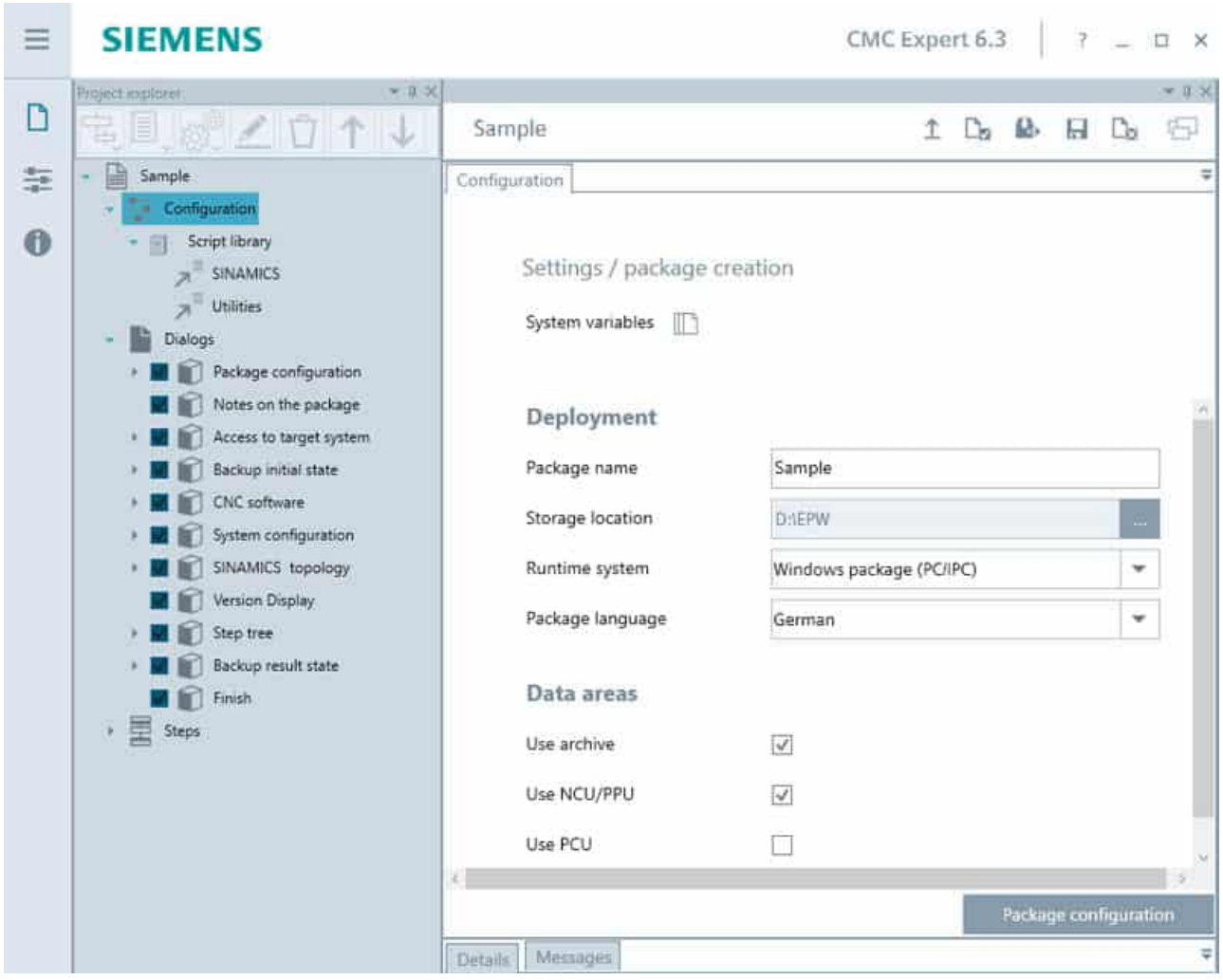


Figure 8-4 Default Expert user interface

2. You can move the "Project explorer", for example, from the left to the right-hand side of the Expert user interface. Click on the title bar and drag the "Project explorer" viewlet from its current position until the positioning tiles are displayed.

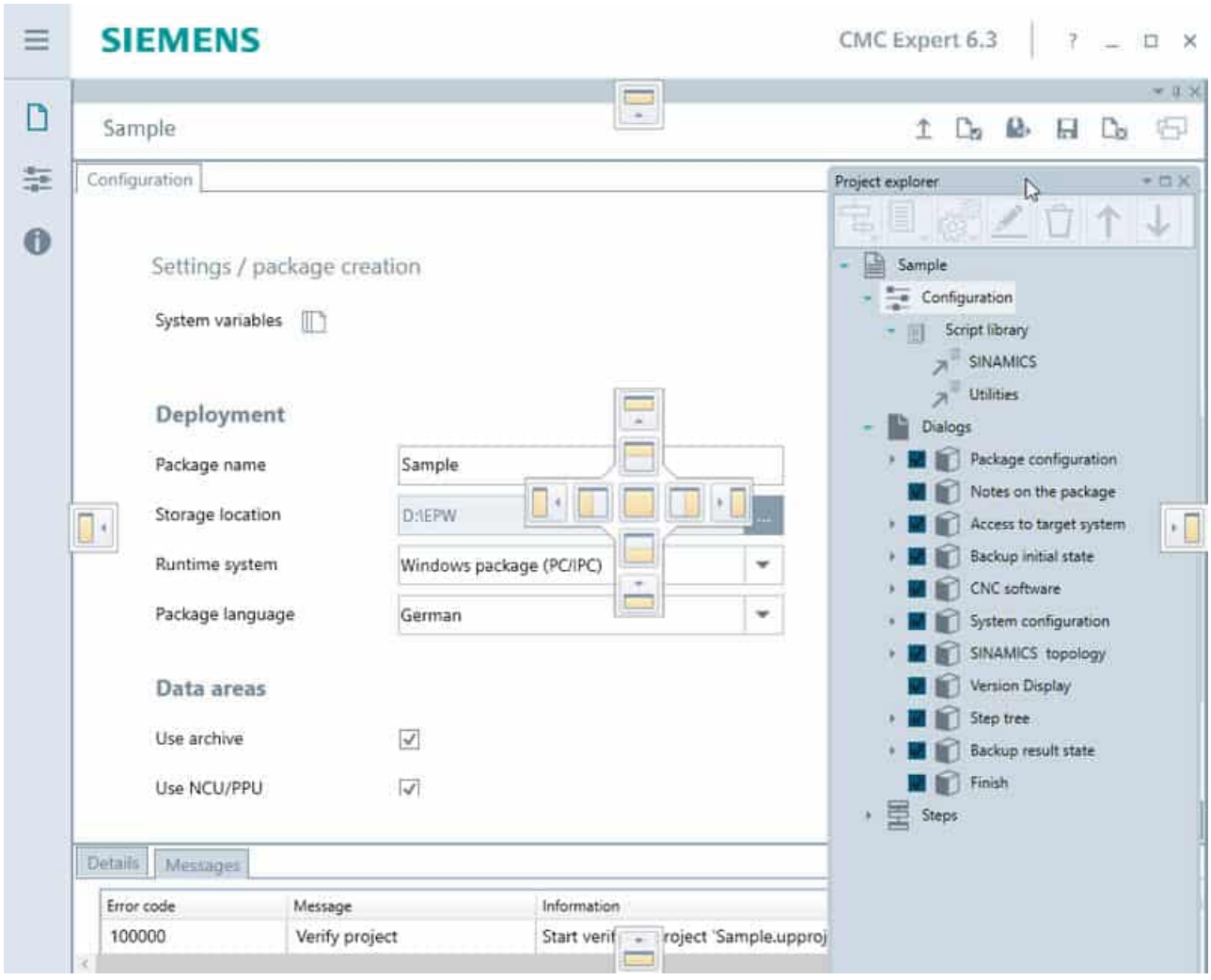


Figure 8-5 Dockable "Project explorer" viewlet

3. Dock the viewlet at the required location using the positioning tile. In this example, the Project explorer is docked at the right-hand positioning tile.
   You can increase or reduce the size of each viewlet to address your specific requirements.

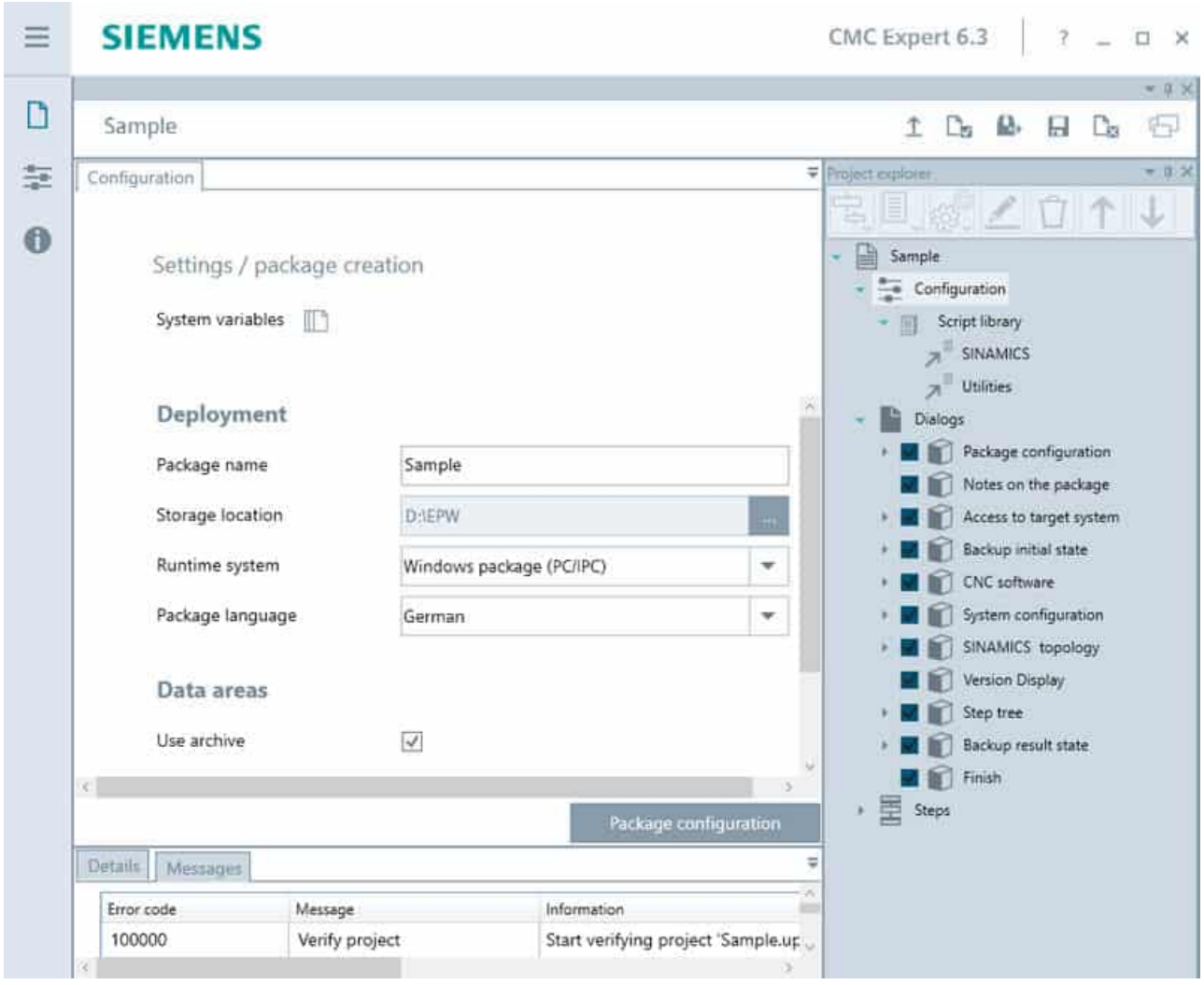


Figure 8-6 Relocating the dockable viewlet using the "Project explorer" as example

4. Another option is that the dockable viewlet is displayed separately from Expert as individual window next to the Expert user interface or on a second screen.
   To do this, drag the required viewlet out of the Expert user interface.

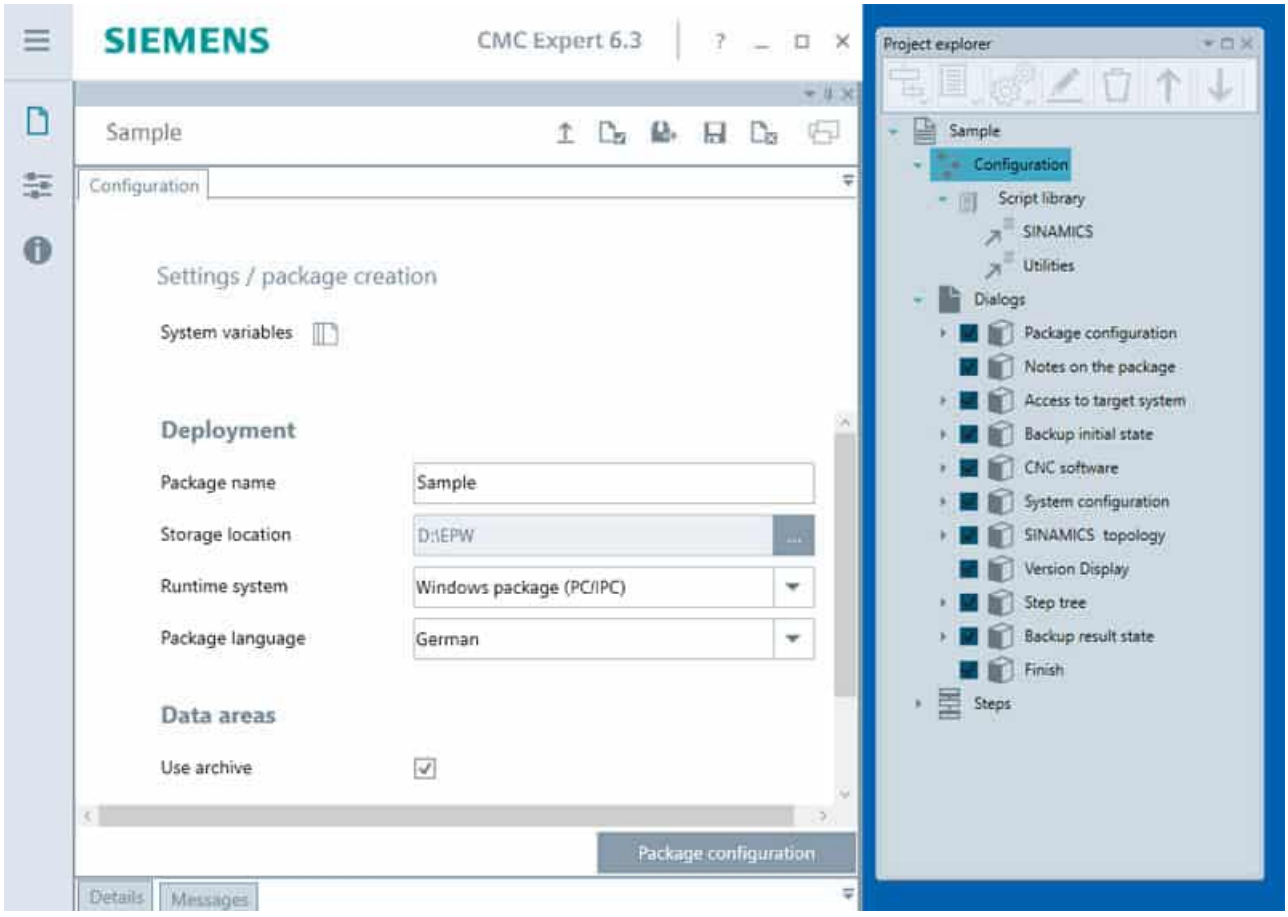


Figure 8-7 Displaying a dockable viewlet separated from Expert as its own window

5. The "Details" and "Messages" windows offer various display options:
   - By default, "Details" / "Messages" are displayed at the bottom of the Home area
   - You can dock the details and messages as tabs in the Home area
   - Automatic display in the background
   - As a separate window next to the Expert interface or on a second screen

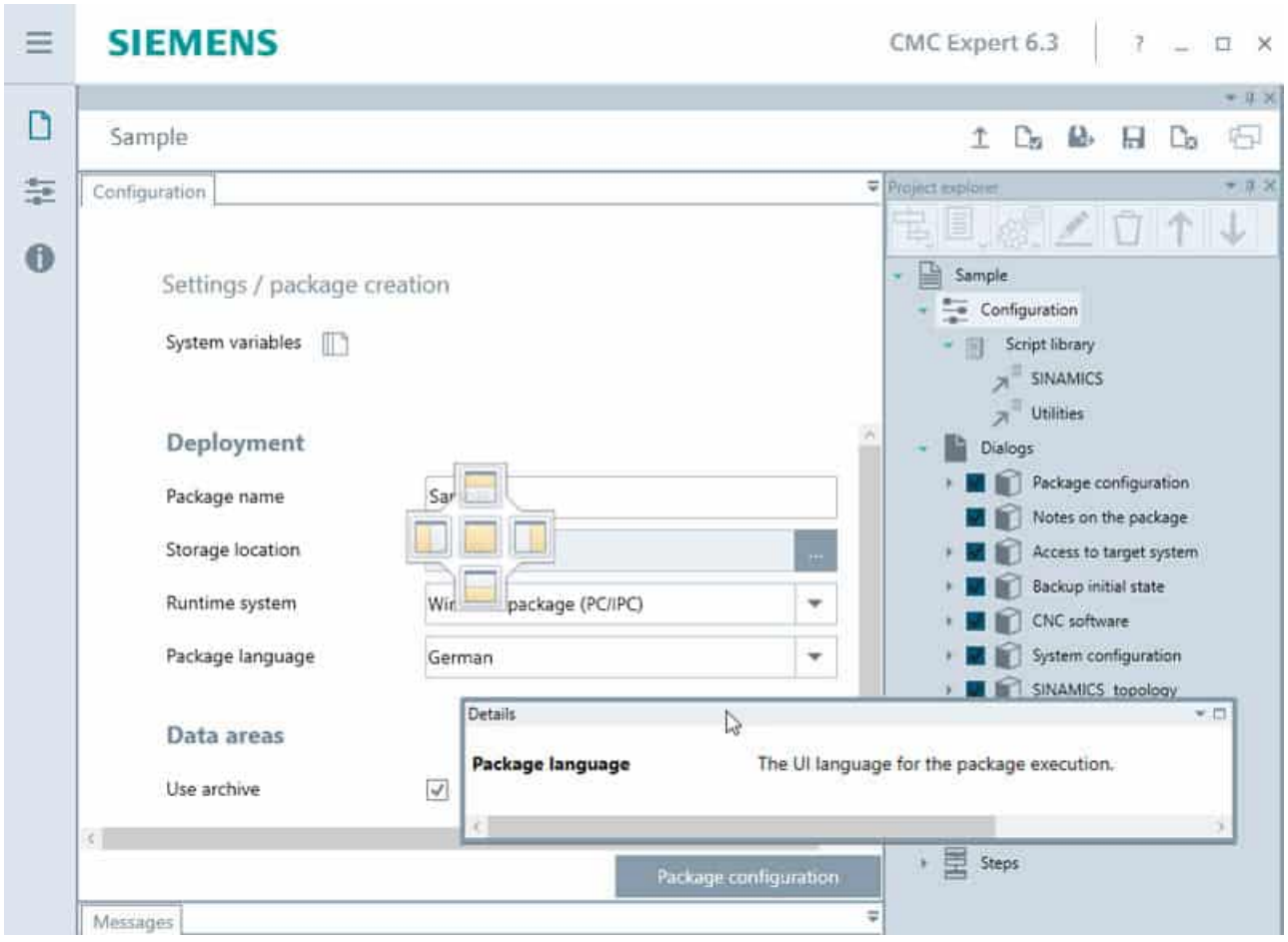


Figure 8-8 Display options of the "Details" or "Messages" windows

8.1.3 Check run

A "check run" is integrated in Expert and shows the errors in the configuration.

Note

The check run does not completely identify syntax errors.

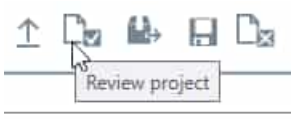
Expressions with the replacement operator are not checked for correct syntax.

Note

Axis-drive assignment

An axis-drive assignment is not checked for completeness. The topology is thus also passed on without any assignment.

Activating the check run



You start the check run of a project in the "Home area" viewlet in the toolbar via "Review project".

By default, the display of the "check run" is located in the "Messages" viewlet under the "Home area" viewlet of Expert.

The display options for the "Messages" viewlet can be found in the "User-friendly Expert interface (Page 223)" section.

8.1.4 Opening a project from CMC Version 6.0

An Expert project as of V6.0 can be opened, saved and further edited with CMC Version V6.1.

For an expert project from V6.1, you must confirm a message, the project is converted and can then be opened and further processed/edited.

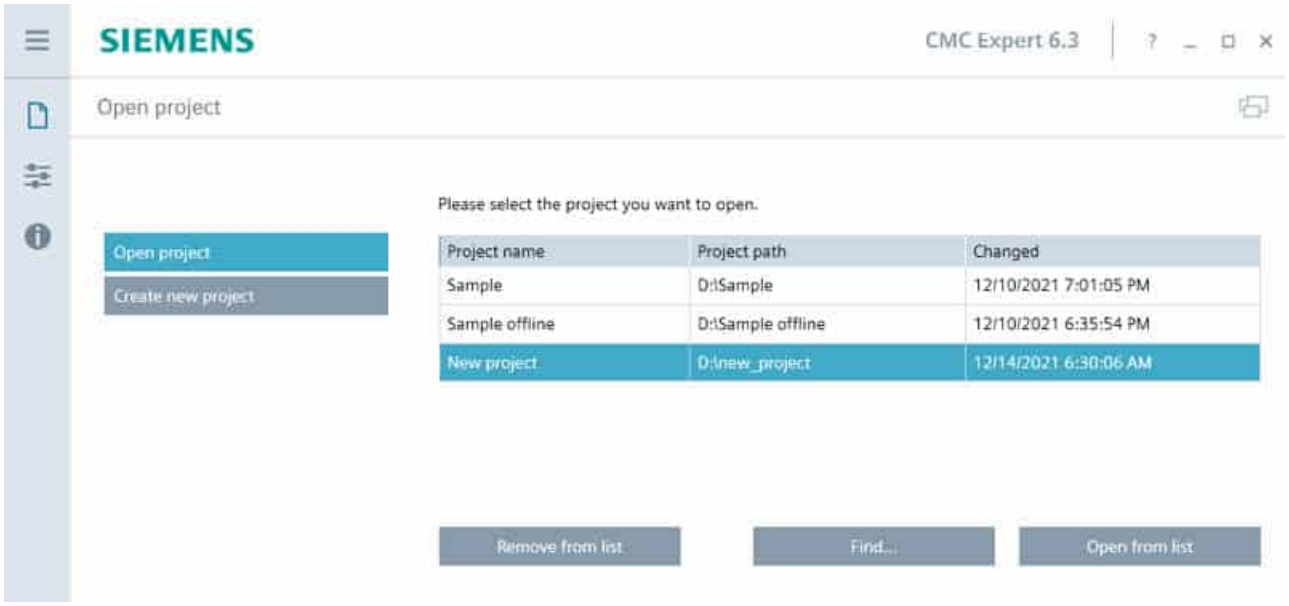
We recommend that you manually back up or archive each project with all of its internal and external components before opening it with the higher CMC version.

8.2 Creating/configuring a project

8.2.1 Opening Expert

Open the Expert configuration software via:

- The Expert icon on the desktop
- Start menu of Windows via input: Create MyConfig Expert 6.3



In the opened Expert, select one of the buttons:

- Open project
- Create new project

Open project

Browse... You open an existing project using the "Browse..." button.
Navigate via the path to the desired Expert project "*.upproj", select and open the project file.

Open from list Projects that have already been used are displayed in a list.
Select the desired project in the list and open the project:

- By left-clicking on the "Open from list" button
- By double-clicking on the required project

Remove from list You can remove projects that you no longer need from the displayed list.
To do this, select the project and remove it by left-clicking on the "Remove from list" button.
The project is deleted from the list of used projects without a prompt for confirmation.

Creating a new project

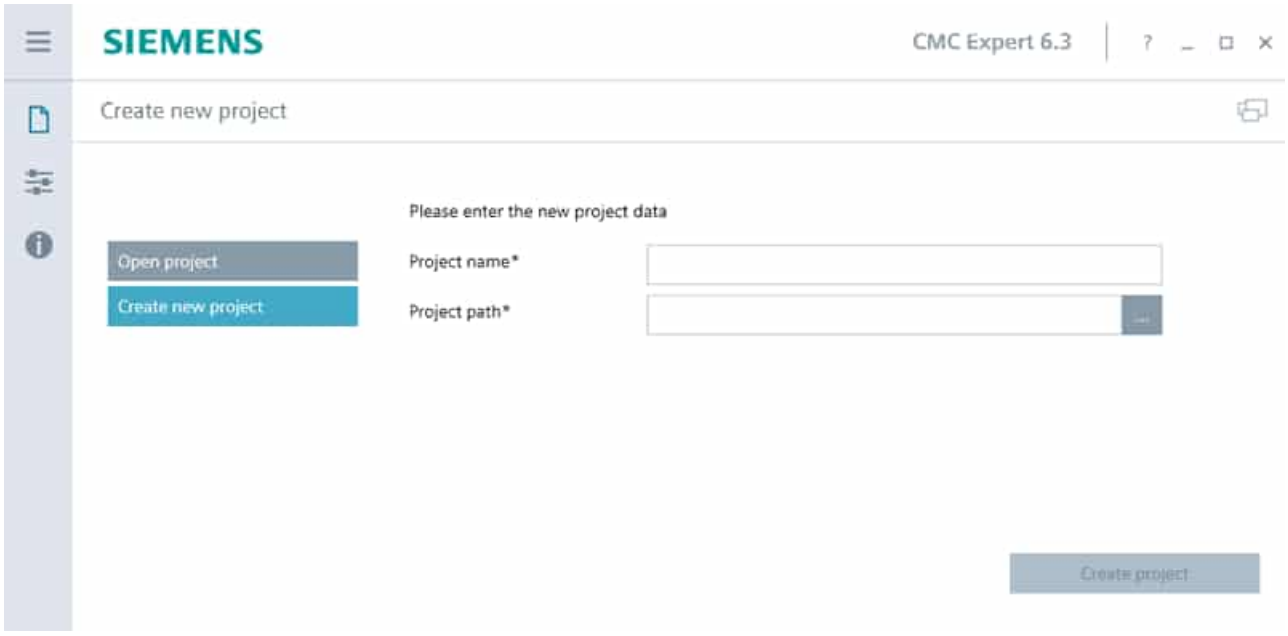
More information can be found at:

Creating a new project (Page 232)

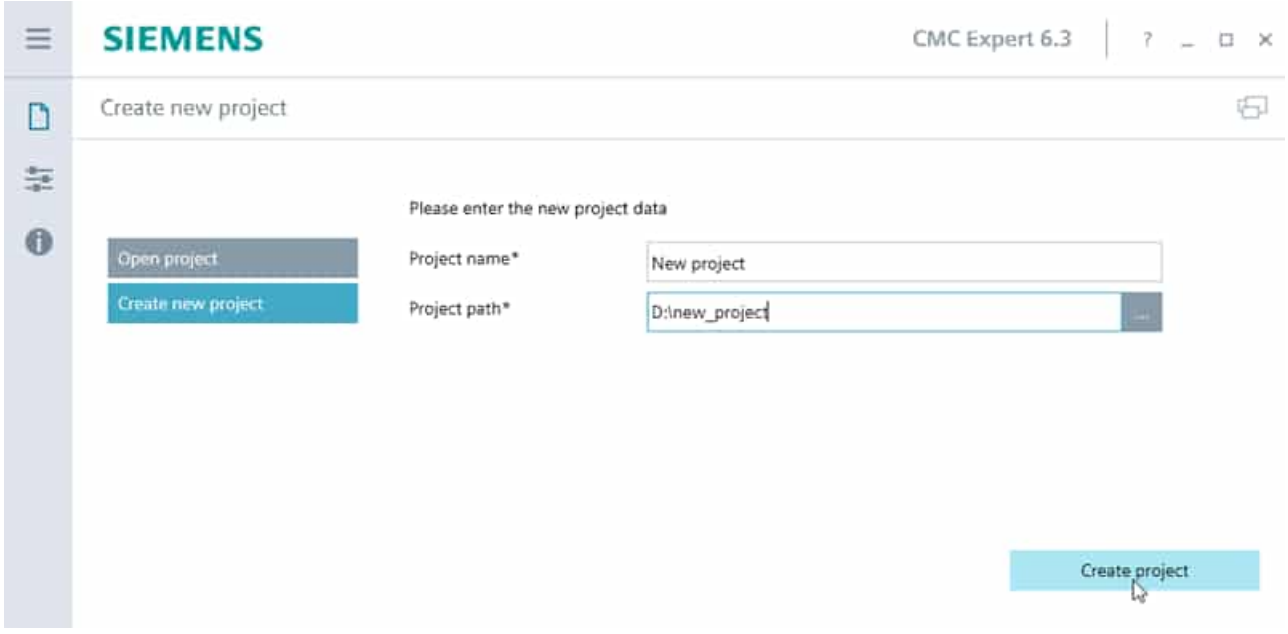
8.2.2 Creating a new project

To create a new project, proceed as follows:

1. In the opened Expert, select the "Create new project" button on the on the entry page.



2. Enter the project name and project path.
   The project directory must be empty.



3. To create and save the new project, click the "Create project" button.
4. The new project with configuration and the dialog pages is opened. The properties of the configuration to be configured are displayed in the home area.

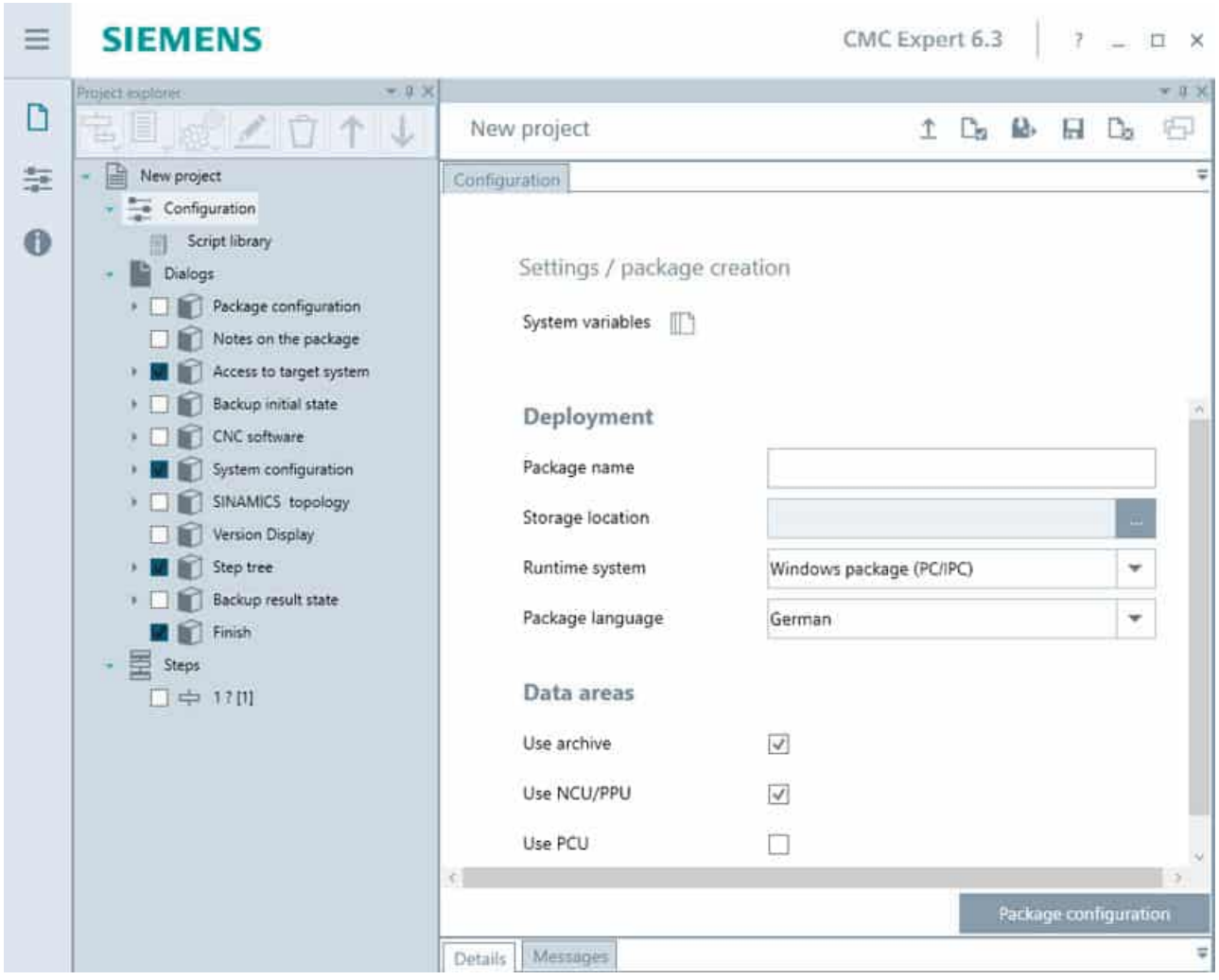


Figure 8-9 Expert - default interface when opening a project for the first time

8.2.3 Project explorer

Structure/view of Project explorer

The "Project explorer" viewlet comprises the following areas:

- Toolbar (Page 216)
- The project tree includes:
  - Configuration
  - Dialog tree
  - Step tree

After you have created a new project (Page 232), the new project is automatically displayed in the center area of the Expert interface in the "Project explorer" viewlet.

The following are displayed in the "Project explorer":

- Configuration (Page 236) for the package execution
- "Dialogs (Page 239)":
  Below the dialogs, the events are already created for which you can create any number of scripts in each case.
  Depending on the set data areas (Archive, NCU/PPU, PCU) in the "Configuration (Page 236)", the following dialog pages are displayed:
  - Package configuration
  - Notes on the package
  - Select offline archive
  - Access to target system
  - Backup initial state
  - PCU setup
  - CNC software
  - System configuration
  - SINAMICS topology
  - Version Display
  - Step tree
  - Backup result state
  - Finish
- "Steps (Page 326)"
  Here you create steps/substeps, scripts and actions. Configure the data in the integrated "Script Editor".

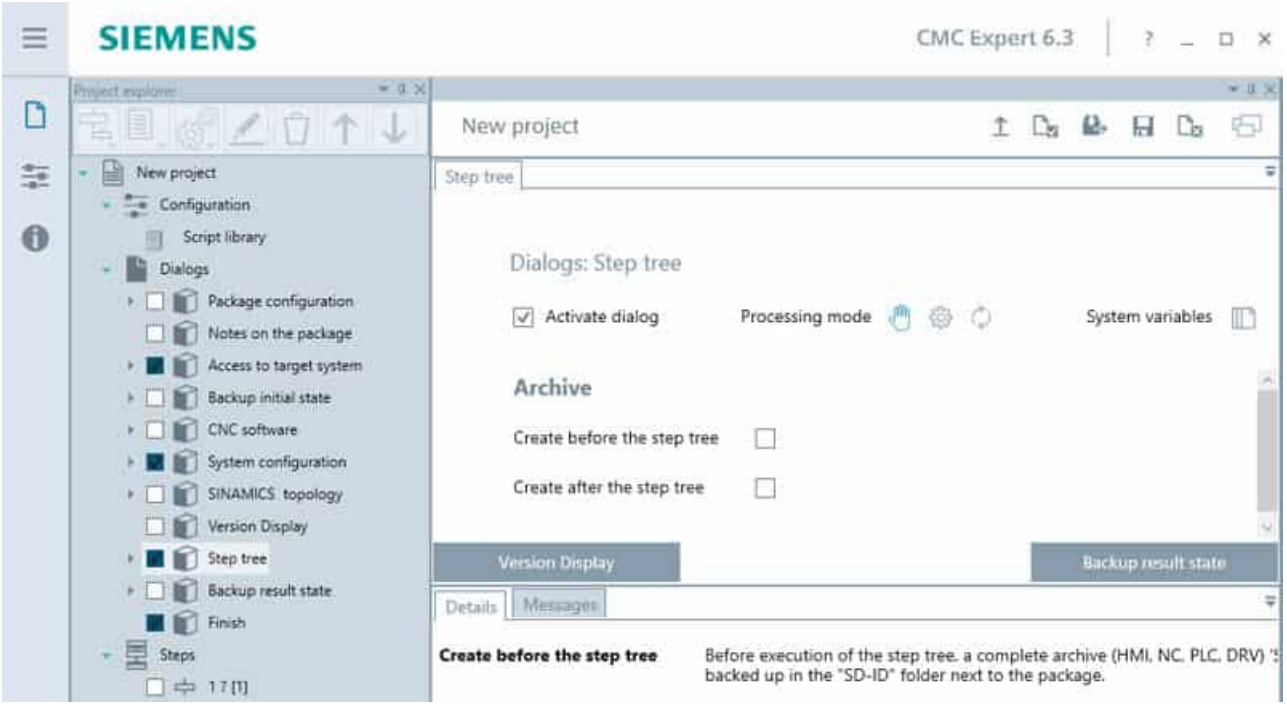


Figure 8-10 Expert project - default representation

Note

Display of properties and details descriptions

The properties for the dialog pages are located on the right-hand side in the Home area viewlet.

When you click on a property, the description for this is displayed below the Home area in the "Details" viewlet.

Systematically configure the data from the top to the bottom. Depending on your configuration, the dialogs will be displayed.

Note

The "Project explorer" viewlet is not displayed

If you inadvertently close the "Project explorer" viewlet, it will no longer be displayed.

Save the project and then reopen it. When the project is reopened, the viewlet is displayed again.

8.2.4 Specifying the configuration for Package

Settings/package creation

In the "Configuration" you define the basic settings for the basic machine and package creation.

This information is passed to the CMC package:

- Deployment
- Data areas

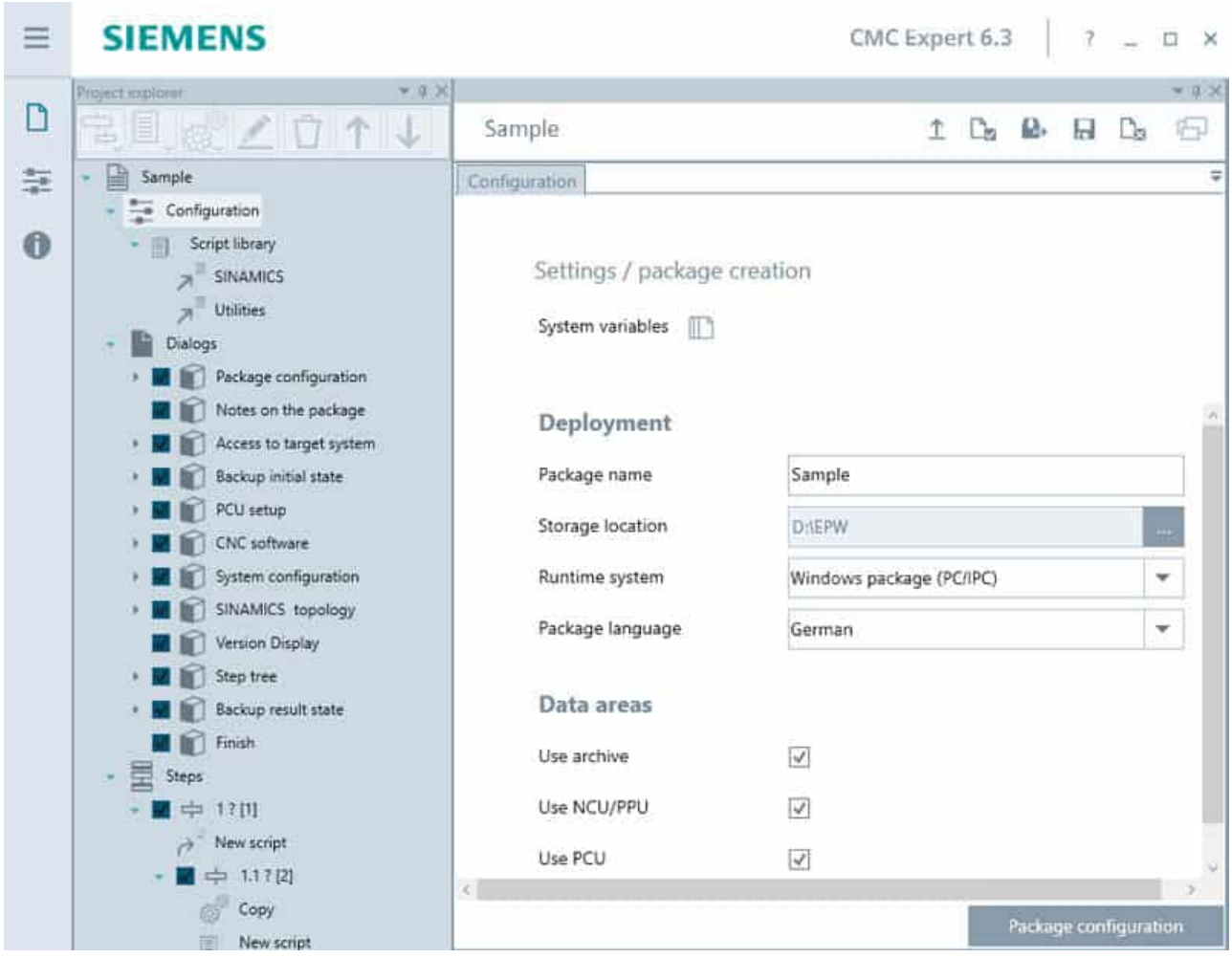


Figure 8-11 Configuration - defining properties

Script libraries

In the Project explorer under "Configuration", you will find the "script library" created by default. From the toolbar, you can link the standard script libraries supplied from the CMC into the script library.

CMC provides you with the CMC standard script libraries:

- SINAMICS.uplip
- Utilities.uplip

The standard libraries are not linked when creating a new project.

To link a library:

1. Select the "Script library" in the Project explorer.
2. In the toolbar, click the "Add script" button.
3. Click on the "Linking an existing script library" button to navigate to the "UpLib" folder.
4. You link the standard libraries from the directory during a standard installation: "c:\Program Files\Siemens\Automation\SINUMERIK\Create MyConfig 6.3\UpLib\"

Note

More information

See the section under "Create MyConfig - CMC script" for more information on the "CMC standard script libraries (Page 39)".

System variables

If you click on the icon next to "System variables", the "System variables for configuration" window is displayed. You can copy the system variables.

More information on the system variables for configuration can be found in the section "System variables for configuration (Page 381)".

Deployment

The following properties are available:

Table 8-5 Configuration > Deployment

Package name		Specify the package name for deployment.
Storage location		Enter the absolute folder path into which the package should be deployed.
Runtime system	Windows package (PC/IPC)	Select the runtime system on which the package should be executed.
	Linux package (NCU)	

Language selection package	German	The project engineer specifies the language selection for the package execution interface. The operator can change the language selection for the package execution. The language is changed using "Ctrl L". The following languages are supported: • German • English
	English	

Data areas

You control the following areas or data by selecting the data areas:

- Configuration of the basic machine
- Selecting the dialog pages
- Excluding data when deploying the package

An overview when selecting dialog pages is provided in Table "Overview depending on the dialog pages of data areas (Page 240)".

The following properties are available:

Table 8-6 Configuration > Data areas

Use archive	☑	With this selection you determine whether the data area "Archive" (NC, PLC, DRV) is used for packet execution. This configuration affects dialogs and scripts. By default, the check box is activated.
Use NCU/PPU	☑	With this selection you define whether data area "NCU/PPU" is used for package execution. This configuration affects dialogs and scripts. By default, the check box is activated.
Use PCU	☐	With this selection you define whether data area "PCU" is used for package execution. This configuration affects dialogs and scripts. By default, the check box is not activated.

8.3 Dialogs

8.3.1 Overview

Overview of dialog pages

In the "Project explorer" in the open project below "Dialogs" , activate or deactivate the dialog pages to be executed and configure their content and behavior.

You can see the dialog list with the dialog pages that can be configured for an example project below:

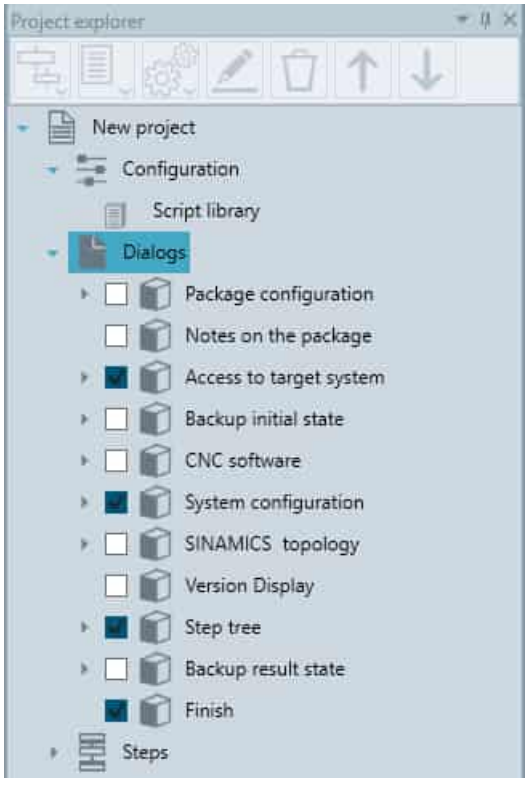


Figure 8-12 Overview of dialog pages

When selecting "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined under "Dialogs".

The dialogs for the package execution are processed from top to bottom in the order in which they are listed.

Non-activated dialogs are ignored for the package execution. The user can also control the type of dialog use via dialog scripts.

You can find an overview for the "Dependencies of the dialog pages of data areas (Page 240)" in the section with the same name.

Dialogs

The following dialog pages are run through chronologically during "Package execution (Page 401)".

The following sections describe the property groups and the setting options for the respective dialog pages.

Finally, the configured dialog page is shown how it is displayed for the operator or commissioning engineer during package execution.

To the dialog pages

The links provide detailed information on the individual dialog pages:

- Package configuration (Page 236)
- Notes on the package (Page 256)
- Access to target system (Page 259)
- Select offline archive (Page 266)
- Backup initial state (Page 273)
- PCU setup
- CNC software (Page 279)
- System configuration (Page 285)
- SINAMICS topology (Page 299)
- Version Display (Page 309)
- Step tree (Page 312)
- Backup result state (Page 318)
- Finish (Page 322)

8.3.2 Dependency of the dialog pages on data areas

In the "Project explorer", in "Configuration (Page 236)" by selecting "Data areas", define which dialog pages are available under "Dialogs" and/or which dialog pages are automatically used.

Dialog pages that are not permitted in the context of the activated data areas are automatically hidden.

You can find an overview of the interrelationship between "Data areas" and "dialog pages" in the following table.

Table 8-7 Overview of the interdependency of the dialog pages on the selected "data areas"

Data areas (Page 236)	**Use archive**	**Yes**	**Yes**	**Yes**	**Yes**	**No**	**No**	**No**	**No**
	Use NCU/PPU	**Yes**	**Yes**	**No**	**No**	**No**	**No**	**Yes**	**Yes**
	Use PCU	**Yes**	**No**	**Yes**	**No**	**No**	**Yes**	**No**	**Yes**
Dialogs (Page 239)	Package configuration	☐	☐	☐	☐	☐	☐	☐	☐
	Notes on the package	☐	☐	☐	☐	☐	☐	☐	☐
	Access to target system	☑	☑	☑	-	-	☑	☑	☑
	Backup initial state	☐	☐	☐	-	-	-	☐	☐
	PCU setup	☐	-	☐	-	-	☐	-	☐
	Select offline archive	-	-	-	☑	-	-	-	-
	CNC software	☐	☐	-	-	-	-	☐	☐
	System configuration	☑	☑	☑	-	-	-	-	-
	SINAMICS topology	☐	☐	☐	☐	-	-	-	-
	Version Display	☐	☐	☐	-	-	☐	☐	☐
	Step tree	☑	☑	☑	☑	☑	☑	☑	☑
	Backup result state	☐	☐	☐	-	-	-	☐	☐
	Finish	☑	☑	☑	☑	☑	☑	☑	☑

☐ The dialog page can be selected.
☑ The dialog page is activated and is used.
☑ The dialog page is automatically used and cannot be deactivated
\- The dialog page is not available.

8.3.3 Dialog scripts

8.3.3.1 Introduction

When creating a new project, the events "OnInit (Page 245)", "OnNext (Page 245)" and "OnEnd (Page 245)" are automatically created for each dialog.

As of CMC version V6.1, you can create any number of scripts for each event.

Note

Exception

No events are created for the following dialogs:

- Notes on the package
- Version Display
- Finish

Events and scripts

You open the events of a dialog by left-clicking on the arrow to the left of the checkbox of the dialog.

Below an event, you create the desired scripts for the event. Select the event to add a script via the toolbar. You can also rename the script name at any time via the toolbar.

1. Open the events of a dialog.

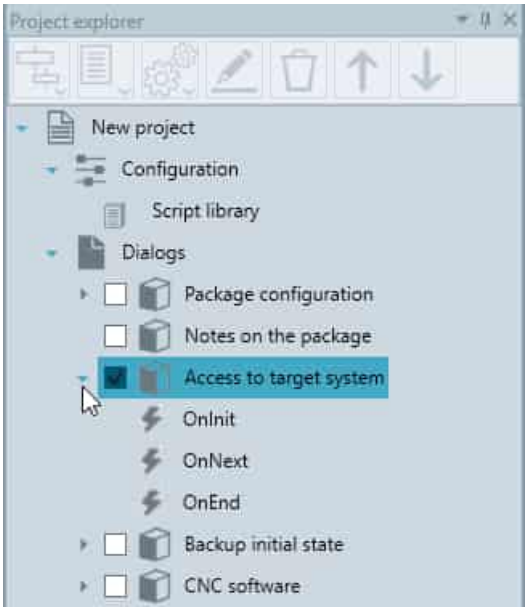


Figure 8-13 Opening events of a dialog

2. Create a new script via the toolbar for a previously selected event.

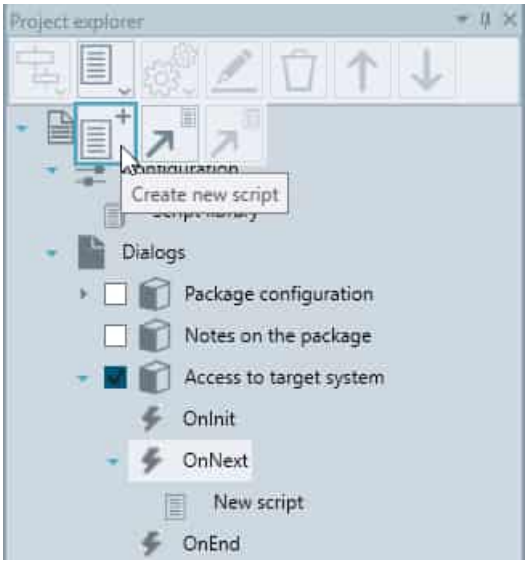


Figure 8-14 Creating a new script

3. To rename a script, select the desired script and click the "Rename" button of the toolbar. The cursor is now located in the text box and you enter the new name.

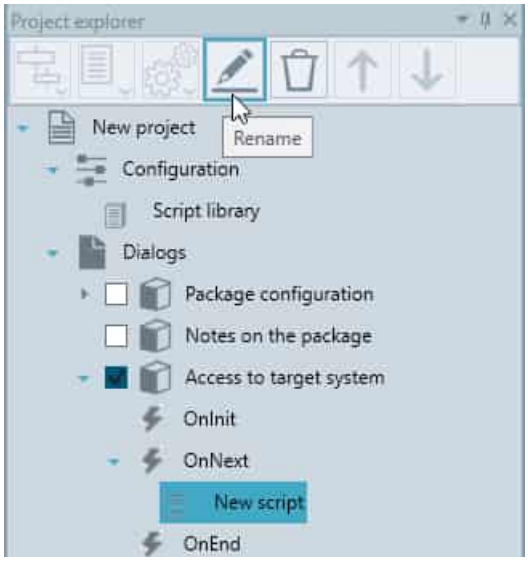
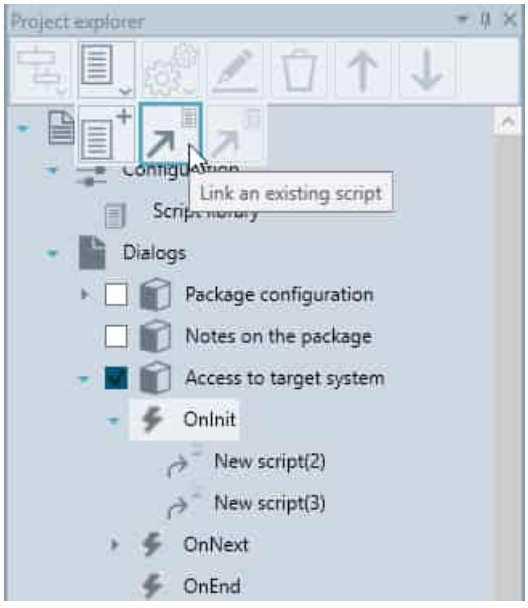


Figure 8-15 Renaming a script

Linking an existing script

You link previously created scripts as follows:

1. Select the event into which you want to link a script.
2. In the toolbar, select "Add script" > "Link an existing script".



3. Navigate to the required script in the open window.
4. Confirm with "Open". The script of the type script file (*.upscr) you selected will be linked and added to the previously selected event.

8.3.3.2 Meaning of the events

The following table describes what you can achieve by using a specific event.

Table 8-8 Event - Meaning

Event	Meaning / Time of script execution
OnInit	The script is executed before the dialog page is displayed. **OnInit** is used, for example: • To (conditionally) skip this dialog page • To issue queries to the operator and process them before the dialog page opens • To change settings of this dialog page via system variables based on the current execution
OnNext	The script is executed in the dialog immediately after clicking "Continue". **OnNext** is used, for example: • To check operator inputs after clicking the "Continue" key • To return to the input At this point in time, set values of the dialog page are checked by exporting system variables. You can interrupt the dialog execution in the script by applying the "Redo() (Page 98)" procedure. In this case, you can correct settings of the dialog page. You can restart execution of the dialog page. The "OnNext" script is also executed again.
OnEnd	After processing has been completed, the script is executed at the end of a dialog page. Through write access to system variables you set which of the following dialogs are to be executed or not executed.

The events are shown schematically in the graphic "Executing the dialog pages - schematic representation (Page 246)".

8.3.3.3 Executing scripts in the dialog pages

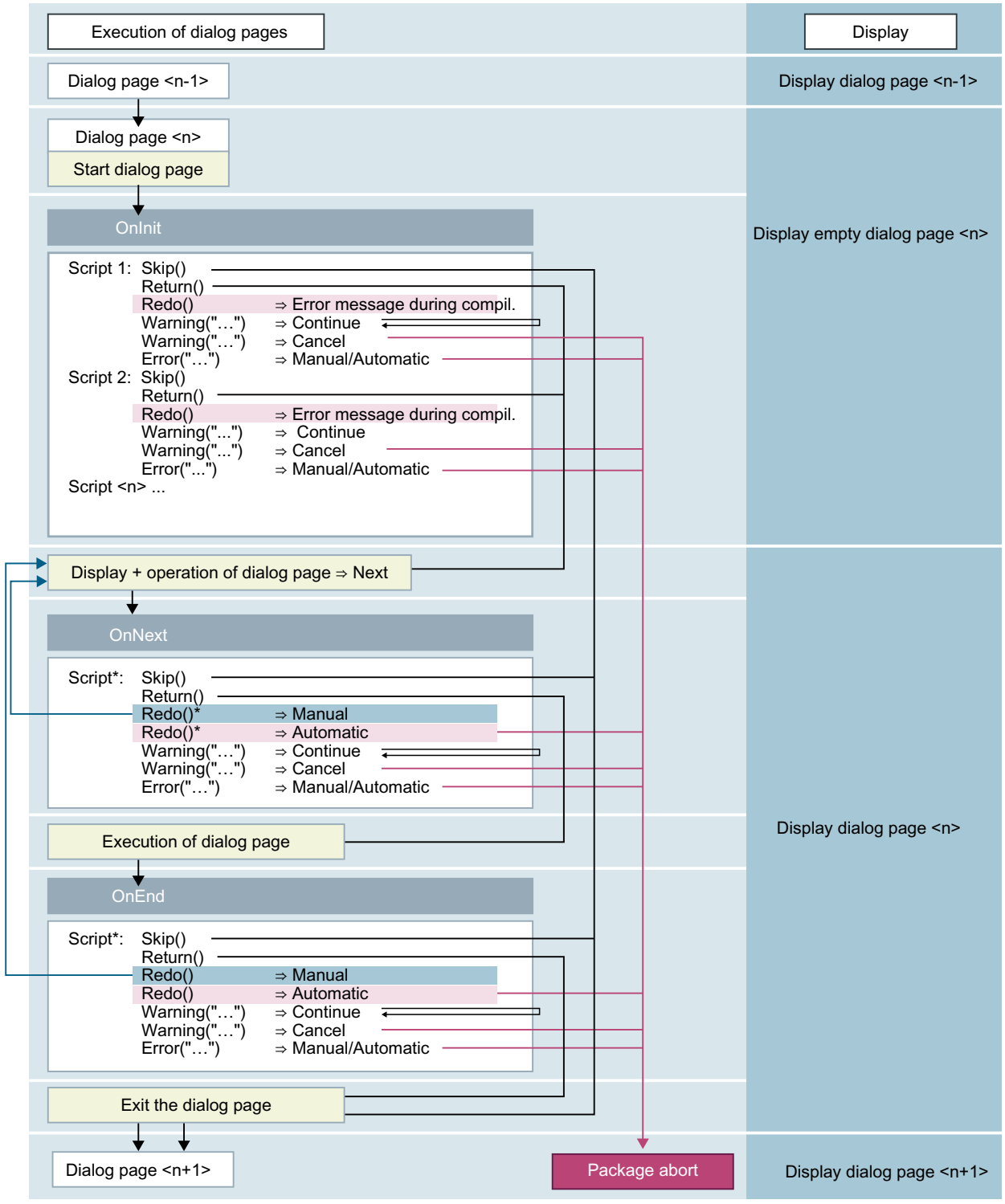
The following graphic shows the execution of the dialog pages and the possible dialog scripts schematically.

The procedures that can potentially control the sequence of the dialog or script are listed as example.

In the dialog script, you can in principle use any of the elements of the script language.

The context menu supports you when generating your script.

More information on the script language can be found in the section or in the online help "Create MyConfig CMC script (Page 29)", under consideration of the constraints in the "Data access (Page 50)".



* You can create any number of scripts for each event.
Only one script is displayed here.

Figure 8-16 Executing dialog pages - schematic overview

Note

Exceptions: `Skip()` not permissible

It is **not** permissible to use `Skip()` on the following dialog pages and it is rejected with an error message during the check run.

- Select offline archive
- Access to target system
- System configuration

More information

Detailed information on the events can be found under:

- OnInit (Page 247)
- OnNext (Page 248)
- OnEnd (Page 250)

8.3.3.4 OnInit event

OnInit details

You can create any number of scripts for the "OnInit" event.

The script(s) is (are) executed before the dialog page is displayed.

You can control the flow of the dialog with the procedures used as examples.

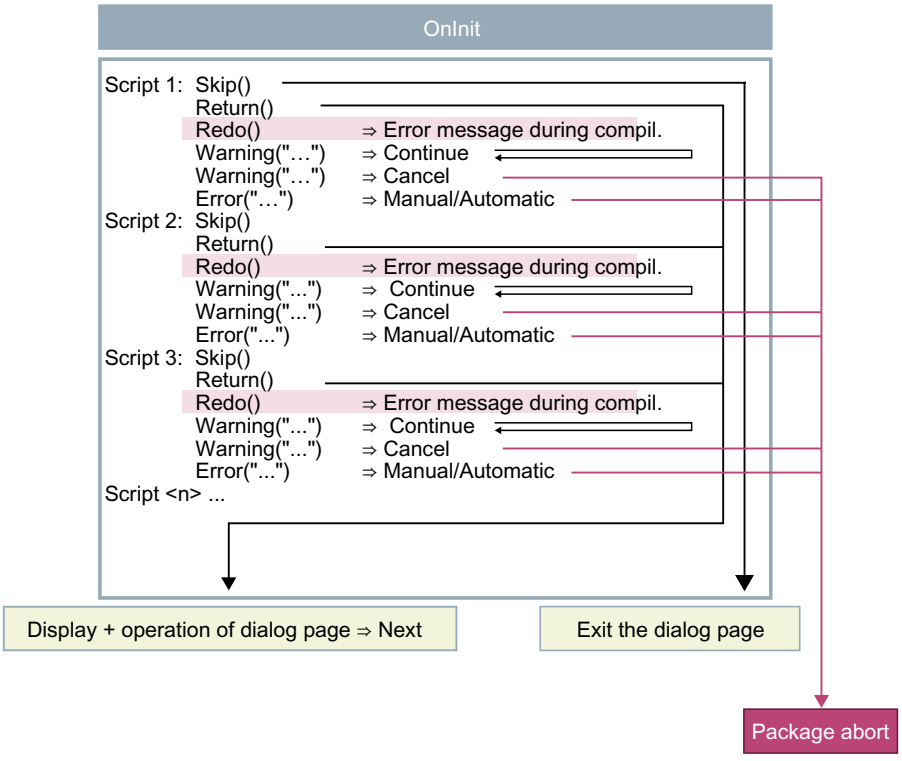


Figure 8-17 "OnInit" event details

Procedure	Description
Skip() (Page 98)	Click `Skip()` to exit the dialog.
Return() (Page 97)	With `Return()` you end the script and either jump to the next available script or, if no script is available, the next dialog is jumped to.
Redo() (Page 98)	`Redo()` causes a termination and an error message in the check run.
Warning() (Page 79)	With `Warning()` the script can be continued after confirmation or `Warning()` leads to cancelling the package.
Error() (Page 79)	`Error()` behaves identically in manual or automatic mode, the package is cancelled.

8.3.3.5 OnNext event

OnNext details

You can create any number of scripts for the "OnNext" event.

The script(s) is (are) executed in the dialog immediately after clicking "Continue".

You can control the flow of the dialog with the procedures used as examples.

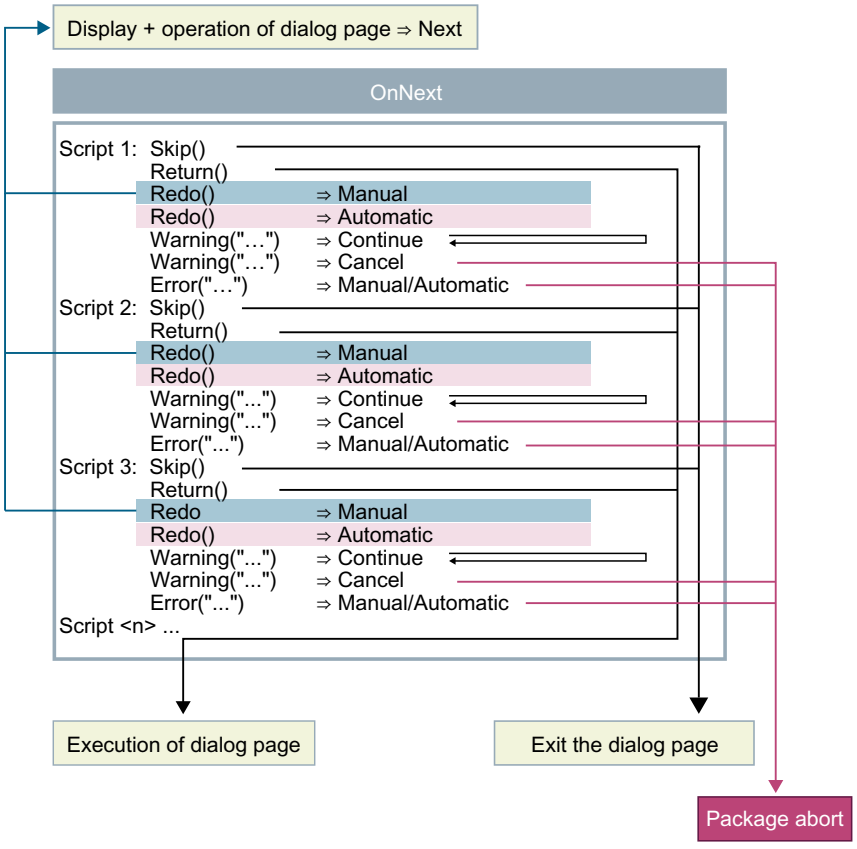


Figure 8-18 "OnNext" event details

Procedure	Description
Skip() (Page 98)	If possible, `Skip()` will jump to the next dialog. The current dialog page is thus no longer executed. If the check run reports errors, the dialog is not skipped.
Return() (Page 97)	With `Return()` you end the script and either jump to the next available script or, if no script is available, the next dialog is jumped to.
Redo() (Page 98)	It behaves differently in automatic sequence than in manual sequence. `Redo()` cancels the automatic sequence. You can pause manually to correct settings of the dialog page. Use Redo() preferably in the dialogs in the "OnNext" event to monitor manual entries.

Procedure	Description
Warning() (Page 79)	With `Warning()` the script can be continued after confirmation or `Warning()` leads to cancelling the package.
Error() (Page 79)	`Error()` behaves identically in manual or automatic mode, the package is cancelled.

8.3.3.6 OnEnd event

OnEnd details

You can create any number of scripts for the "OnEnd" event.

You can control the flow of the dialog with the procedures used as examples.

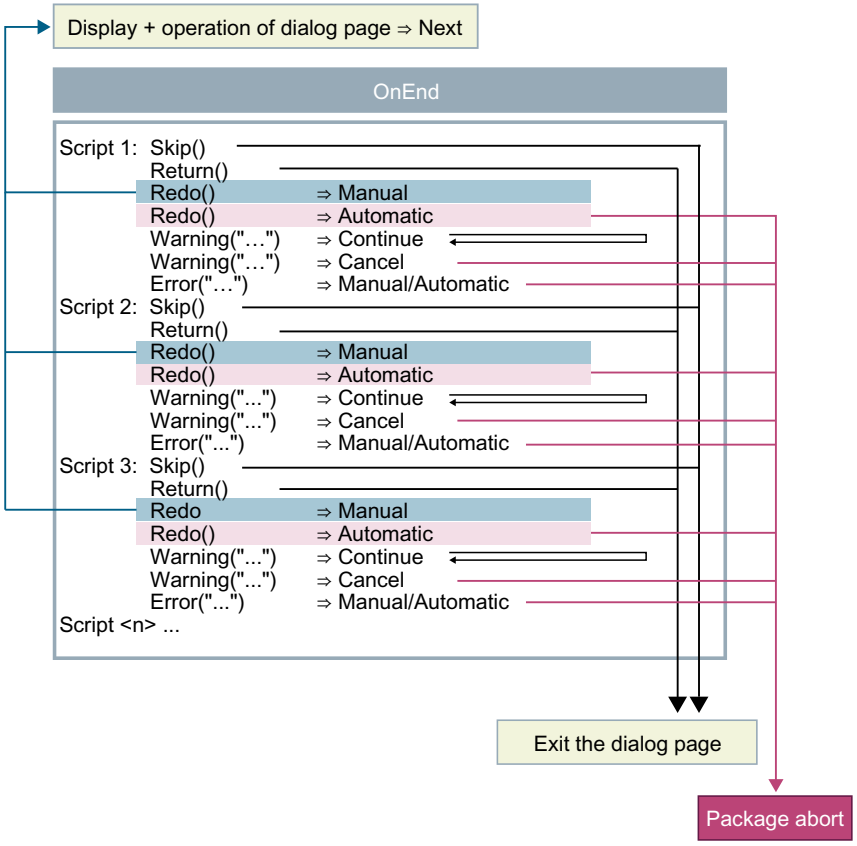


Figure 8-19 "OnEnd" event details

Procedure	Description
Skip() (Page 98)	If possible, `Skip()` will jump to the next script within the dialog or to the next dialog. The current dialog page is thus no longer executed. If the check run reports errors, the dialog is not skipped.
Return() (Page 97)	With `Return()` you end the script and either jump to the next available script or, if no script is available, the next dialog is jumped to.
Redo() (Page 98)	It behaves differently in automatic sequence than in manual sequence. In the automatic sequence, `Redo()` will cancel the package. You can pause manually to correct settings of the dialog page. Use Redo() preferably in the dialogs in the "OnNext" event to monitor manual entries.
Warning() (Page 79)	With `Warning()` the script can be continued after confirmation or `Warning()` leads to cancelling the package.
Error() (Page 79)	`Error()` behaves identically in manual or automatic mode, the package is cancelled.

8.3.3.7 Editing scripts

You edit the dialog scripts in the "Script Editor", which is integrated in CMC Expert.

Open the "Script Editor" in Expert by double clicking on the desired script.

In the home area, a tab opens with the integrated "Script Editor". You can edit new as well as linked scripts.

For example, you can insert CMC scripts via the "Insert script" shortcut menu.

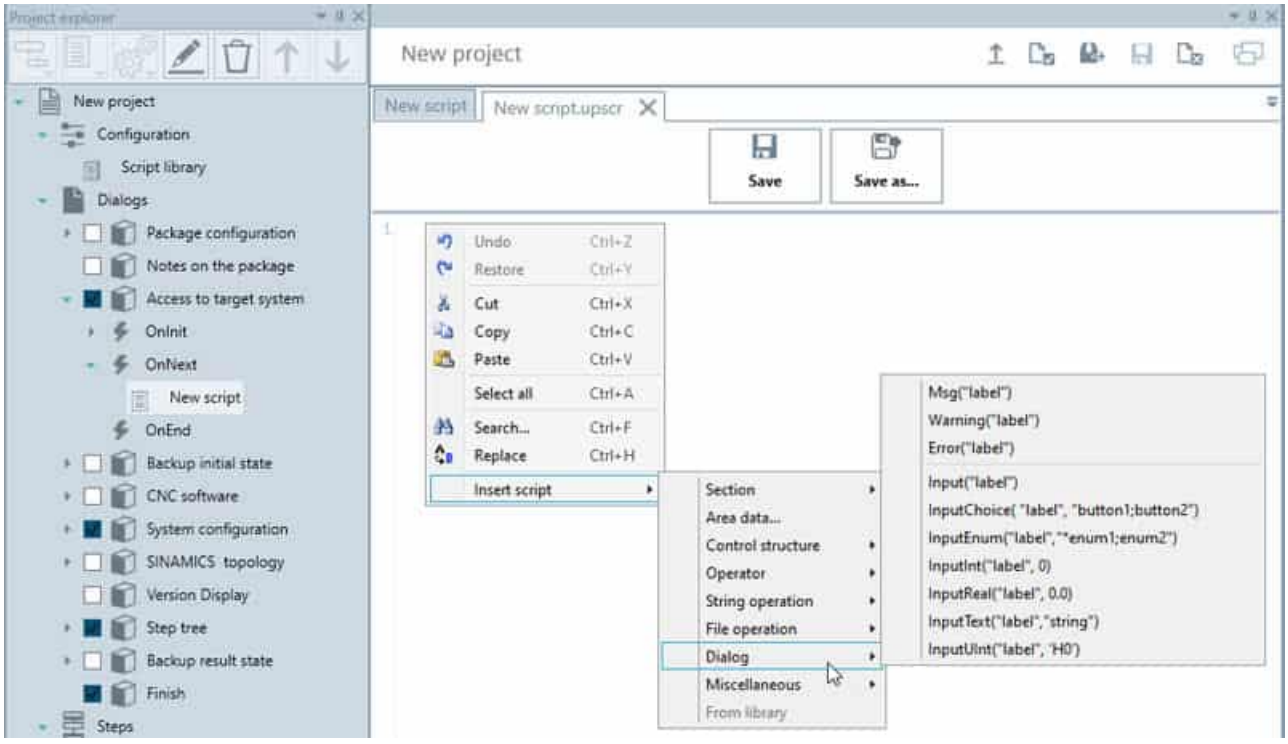


Figure 8-20 Script Editor - editing dialog scripts

Information on inserting a script and on the following options can be found in the section or in the online help "Create MyConfig - CMC script (Page 29)" under:

- Sections (Page 53)
- Area data (Page 49)
- Control structures (Page 54)
- Operators (Page 57)
- String operations (Page 62)
- File operations (Page 65)
- Dialogs (Page 77)
- Miscellaneous (Page 97)
- From library (Page 103)

8.3.4 Package configuration (configuration)

A configuration for executing the package is specified in the "Package configuration" dialog.

Note

Managing configurations

Information on "Managing configurations (Page 350)" can be found in the section with the same name.

You can configure the following property groups for the dialog page "Package configuration":

- Basic dialog settings (Page 220)
- Dialog
- Package configuration (*.upcfg)

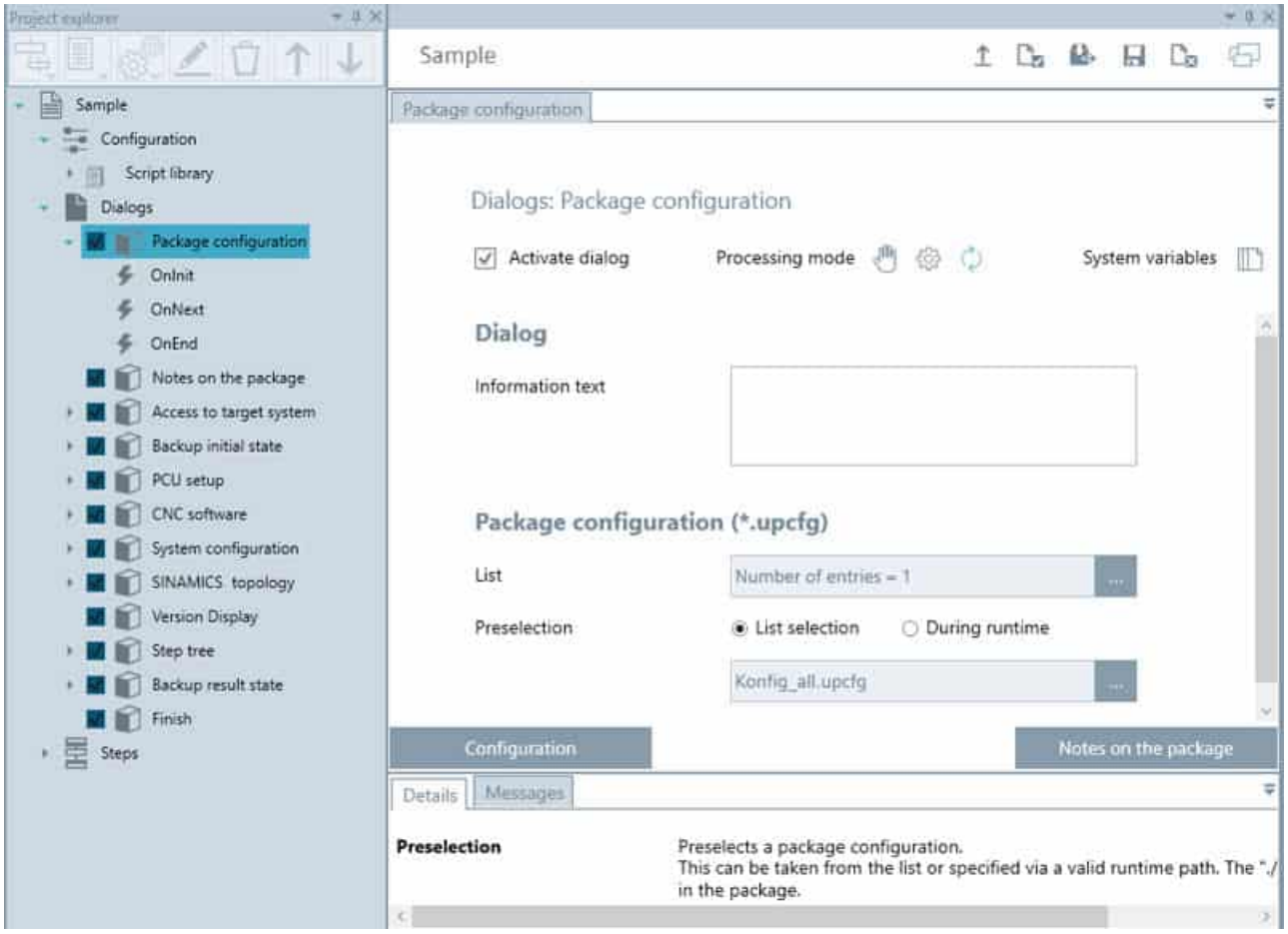


Figure 8-21 Package configuration - project engineer view

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the package configuration dialog (Page 383)".

Dialog

The following setting is available for "Dialog":

Information text	Insert a text for the operation in the field next to "Information text". Formatting with HTML is possible.

Package configuration (*.upcfg)

The following properties are available:

Table 8-9 Package configuration

List	The list of all existing package configurations in the package is displayed in the "List (package configuration (*.upcfg))" dialog. Store and manage a list of files.
Preselection	Select a package configuration. Under "Preselection", you have the options: • List selection You can preselect the stored files as a list. • During runtime The package configuration is selected during package execution. Store an absolute or relative path. Relative paths start with "./" and are relative to the storage location of the package. Absolute or relative paths (not links) are not cancelled until the package execution of Shield. If no preselection has been made for the configuration, then there is no preselection for the package execution. The user can then make a selection from the list of supplied objects.

8.3.5 Package configuration (package execution)

A configuration for executing the package is specified in the "Package configuration" dialog.

Dialog view during package execution (for commissioning engineers / operators)

More information on package execution can be found in Section "Package execution (Page 401)".

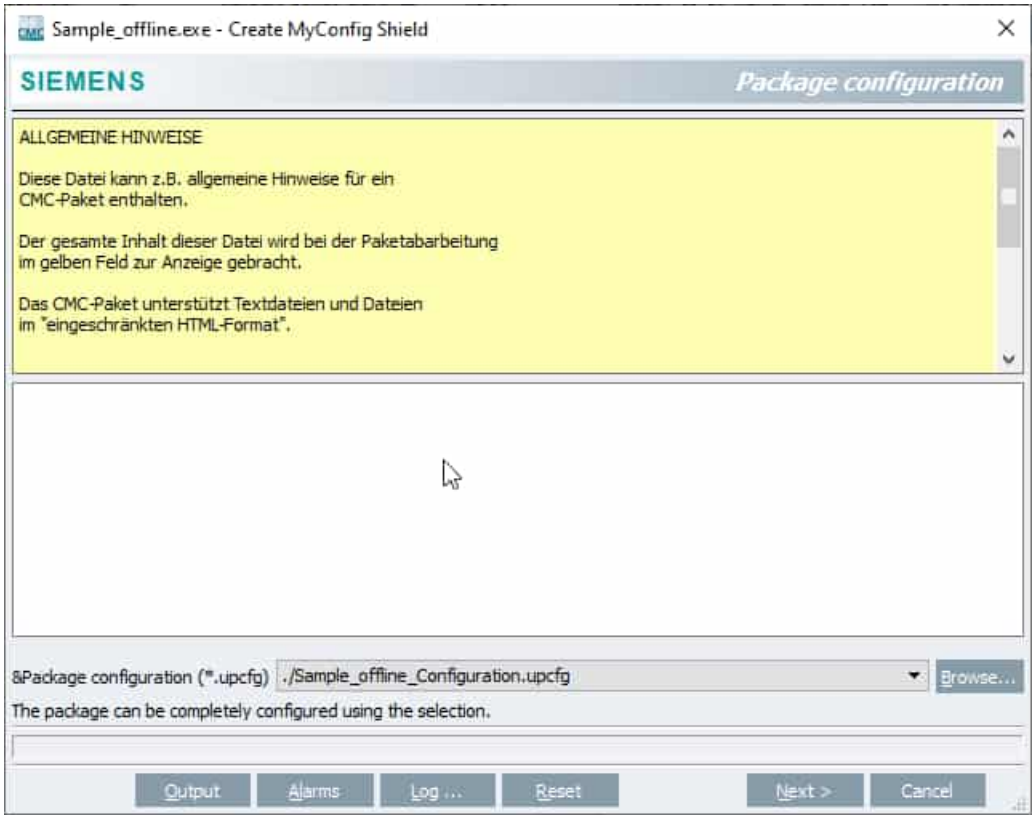


Figure 8-22 Package configuration - commissioning engineer / operator view

Buttons for the package execution dialogs

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

Table 8-10 Package execution - Meaning of the buttons

Button	Meaning
Output	The output window is shown or hidden. Status messages are displayed in the output window while the package is being executed.
Alarms	You can show or hide the alarm list. Errors and messages from NC, PLC, drive and HMI are displayed. The alarm list corresponds to the "Diagnostics" used in SINUMERIK Operate.
Log...	The button opens the CMC logbook for package execution.

Button	Meaning
Reset	You return to starting the package by pressing the "Reset" button.
Continue >	This button takes you to the next dialog page.
Cancel	The package execution is cancelled. In general, a confirmation prompt is displayed that you must acknowledge. After a dialog is canceled, the "Finish" dialog is always displayed, thus allowing the logbook to be backed up. The cancellation process takes place immediately or after a time delay, depending on the dialog page and the execution status.

8.3.6 Notes on the package (configuration)

You communicate general information to the operator via this dialog.

You can configure the following property groups for the "Notes on the package" dialog page:

- Basic dialog settings (Page 220)
- Dialog

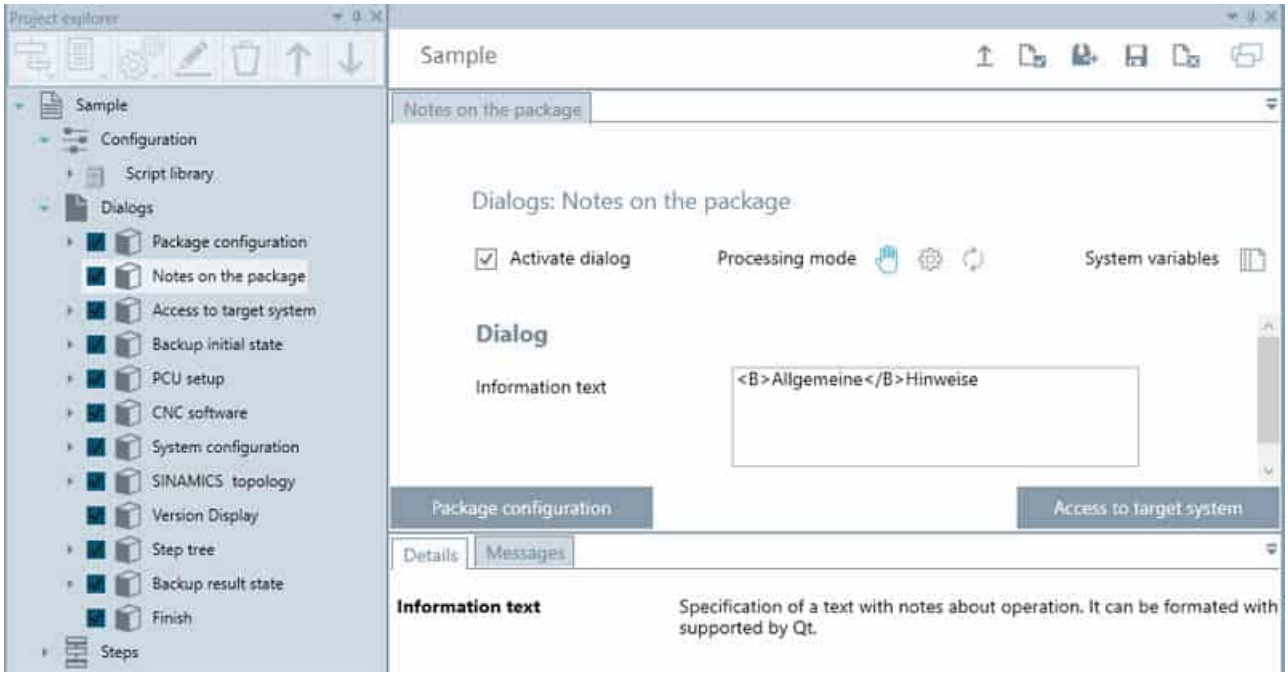


Figure 8-23 Notes on the package - project engineer view

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the Notes on the package dialog (Page 384)".

Dialog

The following setting is available for "Dialog":

Information text	Insert a text for the operation in the field next to "Information text". Formatting with HTML is possible.

Descriptions of properties / buttons

Properties

The properties for the dialog can be found to the right of the "Project explorer" viewlet in the Home area. If you select a property, the description for this is displayed in the "Details" tab as a separate viewlet below the properties.

Buttons

There are buttons in the Home area below the properties:

- Use the left button to navigate back to the previous dialog
- Use the right button to navigate to the following dialog

8.3.7 Notes on the package (package execution)

You communicate general information to the operator via this dialog.

Dialog view during package execution (for commissioning engineers / operators)

A file in text format or HTML format can be displayed with notes about executing the package on the "Notes on the package" dialog page.

More information on package execution can be found in the section under "Package execution (Page 401)".

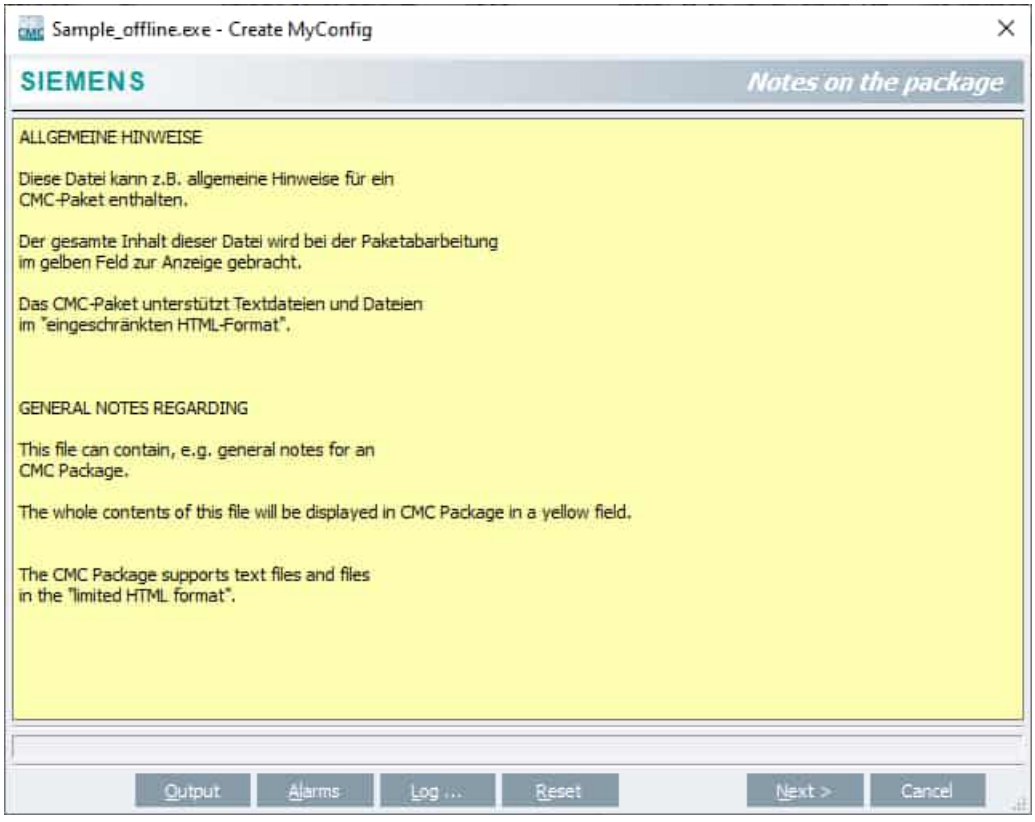


Figure 8-24 Notes on the package - commissioning engineer / operator view

Buttons for the package execution dialogs

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

Table 8-11 Package execution - Meaning of the buttons

Button	Meaning
Output	The output window is shown or hidden. Status messages are displayed in the output window while the package is being executed.
Alarms	You can show or hide the alarm list. Errors and messages from NC, PLC, drive and HMI are displayed. The alarm list corresponds to the "Diagnostics" used in SINUMERIK Operate.
Log...	The button opens the CMC logbook for package execution.
Reset	You return to starting the package by pressing the "Reset" button.

Button	Meaning
Next >	This button takes you to the next dialog page.
Cancel	The package execution is cancelled. In general, a confirmation prompt is displayed that you must acknowledge. After a dialog is canceled, the "Finish" dialog is always displayed, thus allowing the logbook to be backed up. The cancellation process takes place immediately or after a time delay, depending on the dialog page and the execution status.

8.3.8 Access to target system (configuration)

In this dialog, you specify the configuration for executing the package.

You already define the basic configuration and the desired target system in the "Configuration (Page 236)" tab via the activated "Data areas".

For the dialog page "Access to target system", you configure the following property groups depending on the selection of the data areas in the configuration.

Table 8-12 Dependence of target system on selection of data areas (configuration)

Selection of data areas (configuration)			Desired target system	"Property groups" display in the "Access to target system" dialog
Archive	NCU	PCU		
x	x	-	NCU	• Dialog • NCU
-	x	-		
-	-	x	PCU	• Dialog • PCU
-	x	x	PCU	• Dialog • NCU • PCU
x	x	x		

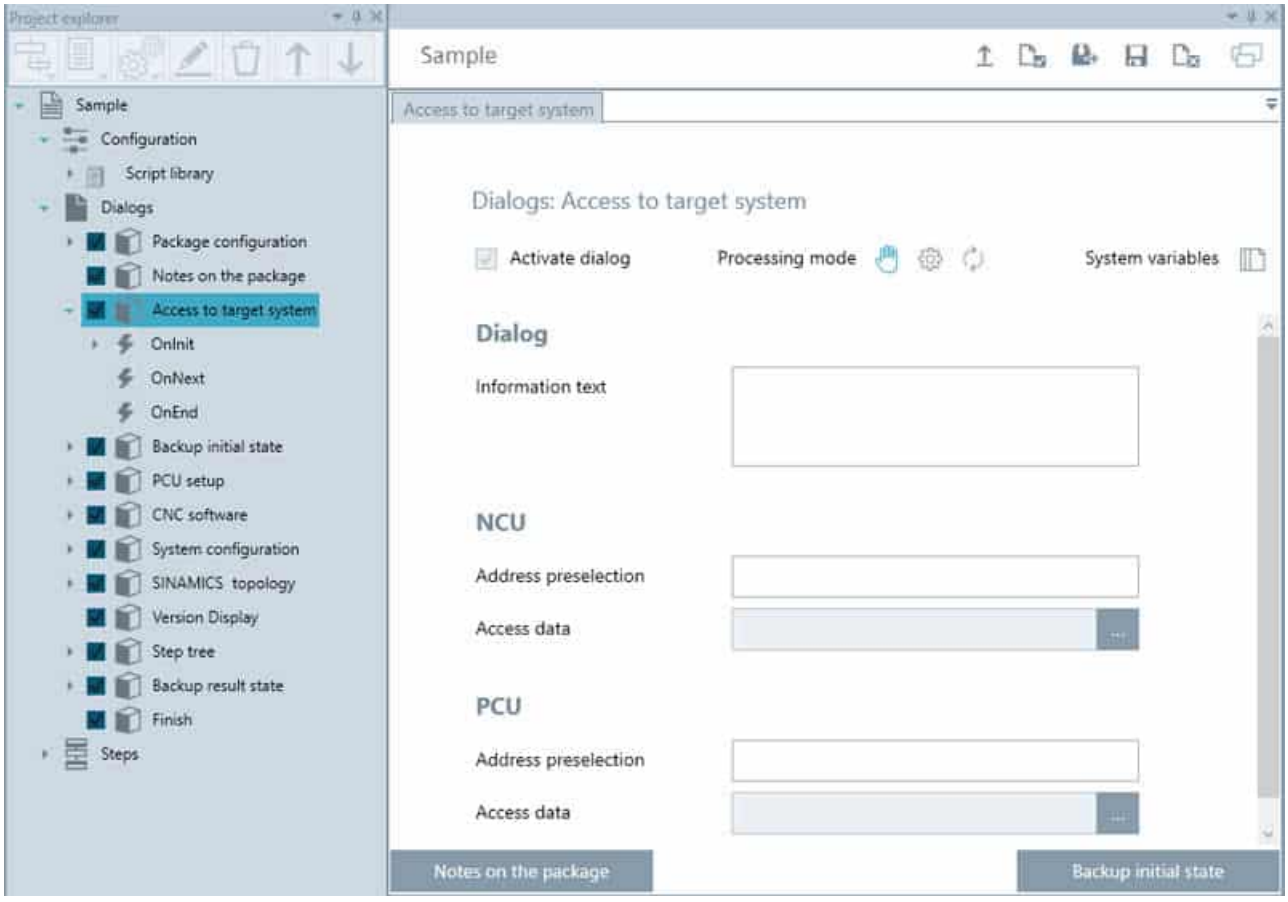


Figure 8-25 Configuration - Access to target system

Note

Remotely accessing the PCU

CMC Version 6.2 or higher also supports the execution of packages locally on the PCU.

CMC 6.3 does not support remote access to the PCU.

Note

Specifying data areas

When selecting the "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined.

You can find an overview of this topic in the section "Dependencies of the dialog pages on data areas (Page 240)".

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the Access to target system dialog (Page 384)".

Dialog

The following setting is available for "Dialog":

Table 8-13 "Dialogs" > Property group "Dialog"

Information text	Insert a text for the operation in the field next to "Information text". Formatting with HTML is possible.

NCU

For "NCU", the following settings are available:

Address preselection	Preselects the IP address of the control (NCU or PPU). The preselection is relevant only if the package is executed on a Windows system. For a connection via "X120/X127", "192.168.214.1" is always available in the drop-down list.
Access data	Click the button to open the "Assign access data" dialog. Enter the following: • User name Specify the user name for an account with manufacturer rights for access to the NCU. Generally you use "manufact". • Password Enter your password for the user name. **Note** Note that the user is responsible for entering a password. • Confirm password The data is backed up via the "Back up access data" button. You can cancel the process via "Cancel".

Note

Package execution and access rights

- For the automatic package execution locally on the "NCU", e.g. via a USB flash drive, the package is automatically assigned the required rights by the system.

PCU

For "PCU", the following settings are available:

Address preselection	Preselect the computer name or the IP address of the PCU. Store an IP address, a computer name or a path for the simulation. The preselection is relevant only if the package is executed on a remote Windows system. For a connection via the system network, "192.168.214.241" is always available in the drop-down list in Shield. **Note** If the package is to be executed locally on a PCU, the specified address has no significance.
Access data	Click the button to open the "Assign access data" dialog. Enter the following: • User name Specify a user name for a user account with administrative rights for remote access to the PCU. The specification is not required for local execution of the package. • Password Enter your password for the user name. **Note** Note that the user is responsible for entering a password. • Confirm password Confirm the password by entering it repeatedly. The data is backed up via the "Back up access data" button. You can cancel the process via "Cancel".

NC access

For "NC access", the following setting is available:

NC access level	Click the button to open the "Assign access data" dialog. Enter the following: • Password Enter your password for NC access. **Note** Note that the user is responsible for entering a password. • Confirm password The data is backed up via the "Back up access data" button. You can cancel the process via "Cancel".

Descriptions of properties / buttons

Properties

The properties for the dialog can be found to the right of the "Project explorer" viewlet in the Home area viewlet. If you select a property, the description for this is displayed in the "Details" tab as a separate viewlet below the properties.

Buttons

There are buttons in the "Home area" below the properties:

- Use the left button to navigate back to the previous dialog
- Use the right button to navigate to the following dialog

8.3.9 Access to target system (package execution)

A configuration for executing the package is defined in dialog "Access to target system".

Dialog view during package execution (for commissioning engineers / operators)

More information on package execution can be found in the section of the same name "Package execution (Page 401)".

Depending on which target system was defined during the configuration, for the package execution, you see the dialog for "NCU", "NCU and IPC" or "Virtual machine".

NCU

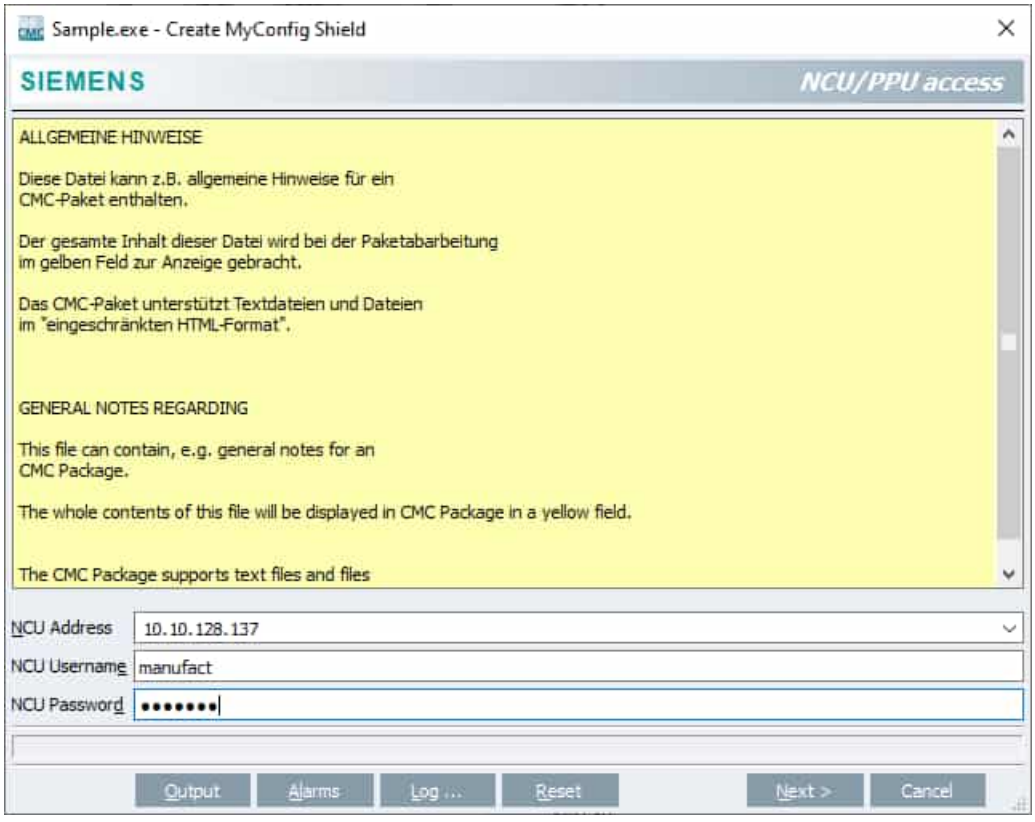


Figure 8-26 Access to target system - NCU

NCU and IPC

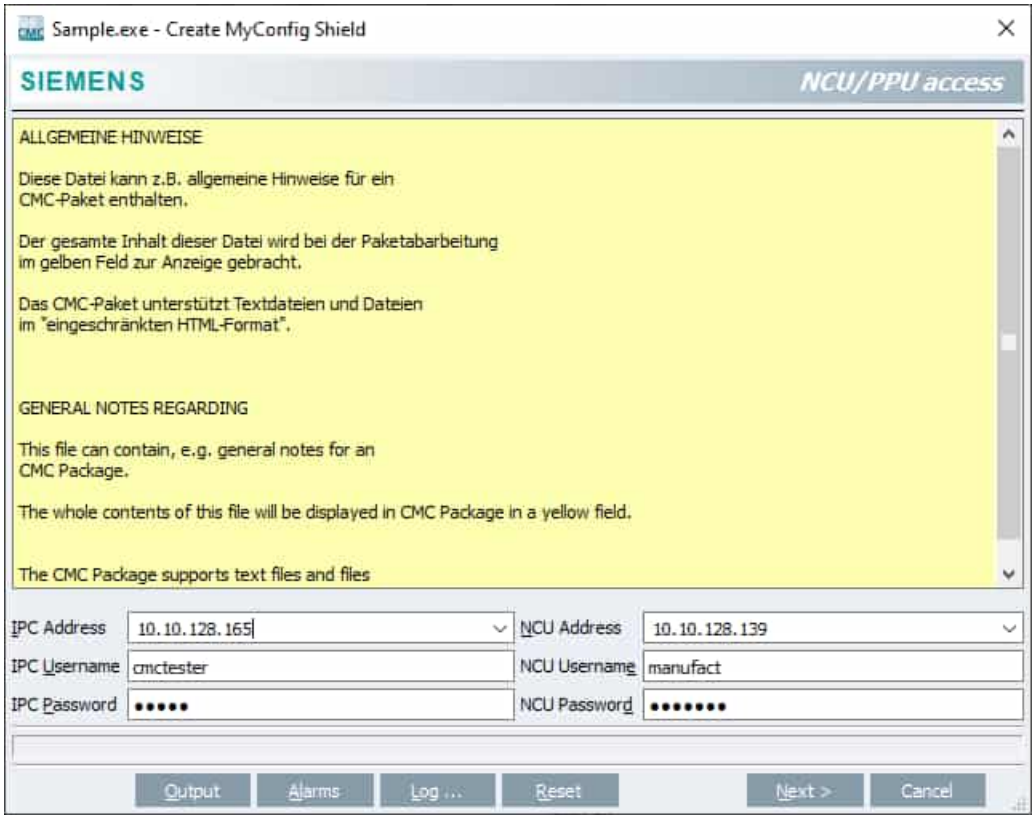


Figure 8-27 Access to target system - NCU and IPC

Virtual machine

When accessing "Virtual machine" from another Windows computer, appropriately populate fields "IPC Adress", "IPC Username" and "IPC Password".

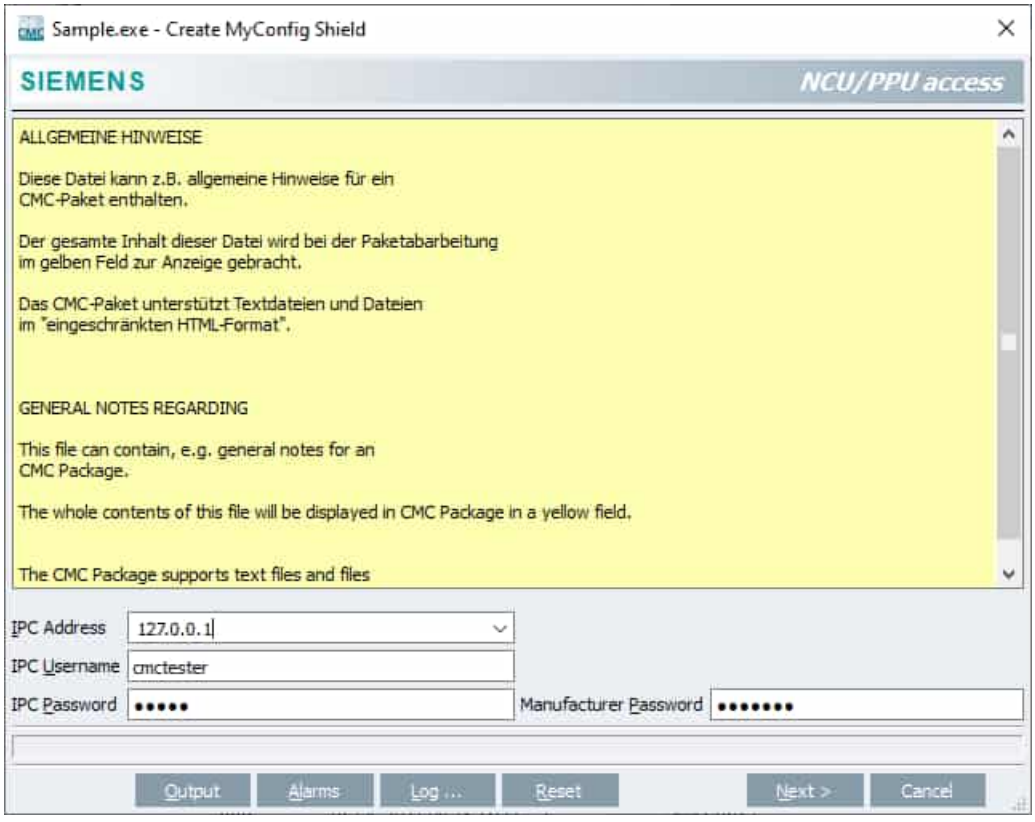


Figure 8-28 Access to target system - "Virtual machine"

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons (Page 258)".

8.3.10 Select offline archive (configuration)

In this dialog, you specify from where the output data for control areas NC, PLC and DRV is obtained for offline package execution without NCU.

You configure the following property groups for the "Select offline archive" dialog page:

- Basic dialog settings (Page 220)
- Dialog
- Offline archives
- Input archive
- HMI archive data
- NC archive data
- PLC archive data
- DRV archive data/drive configuration
- System settings
- Output archive

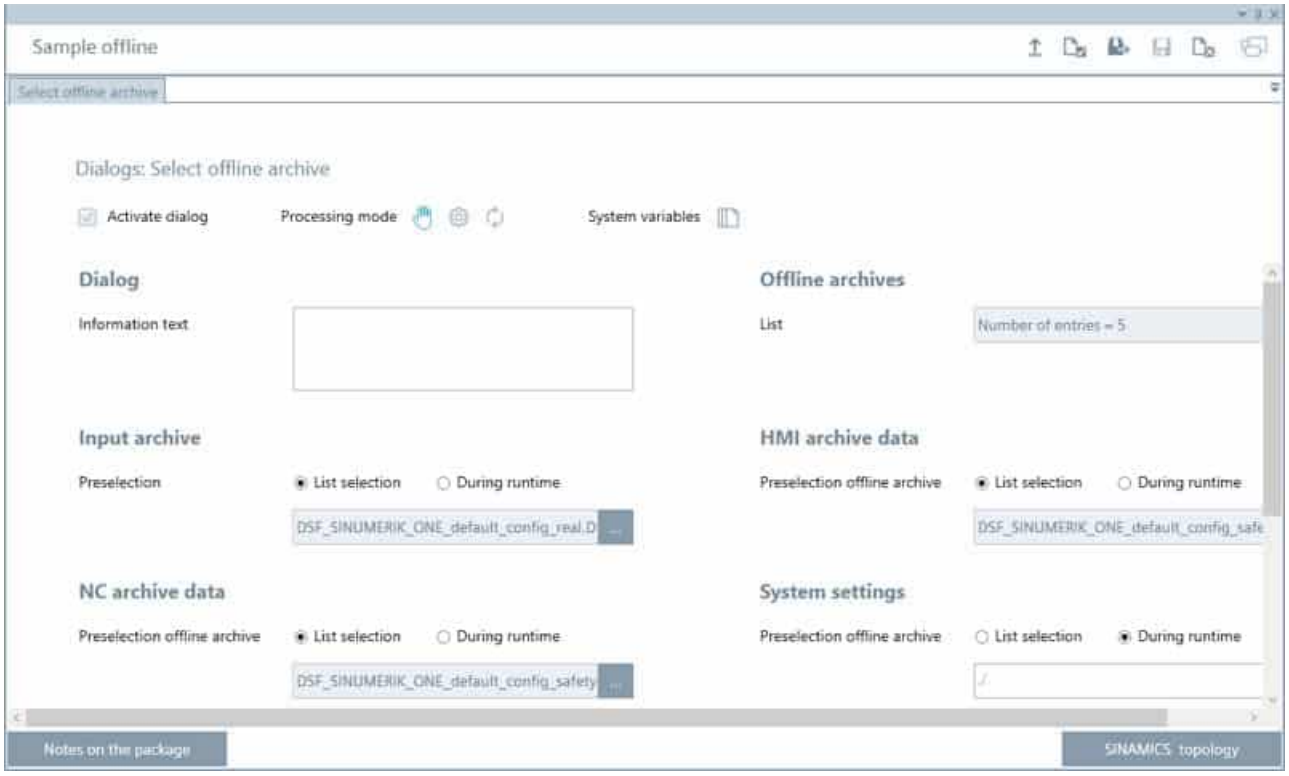


Figure 8-29 Select offline archive - project engineer view

Note

Specifying data areas

When selecting the "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined.

You can find an overview of this topic in the section "Dependencies of the dialog pages on data areas (Page 240)".

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the Select offline archive dialog (Page 385)".

Dialog

The following setting is available for "Dialog":

Table 8-14 "Dialogs" > Property group "Dialog"

Information text	Insert a text for the operation in the field next to "Information text". Formatting with HTML is possible.

Offline archives

The following property is available:

List	Using button , open dialog "List (offline archives)". The list contains the offline archives, which should be transferred together with the package, that you can use for the subsequent preselections and for the package execution.

Input archive

The following property is available:

Preselection	○ List selection ◉ During runtime	Under "Preselection", you have the following options: • List selection You can preselect the stored files as a list. Using button ▪, open dialog "Preselection (input archive)". An archive can be preselected on which all areas contained for package execution are available. Under "Preselection" you can: – Insert a file as link – Select a file from the list – Remove an entry from the list • During runtime The input archive is selected during package execution. Store an absolute or relative path. Relative paths start with "./" and are relative to the storage location of the package. Absolute or relative paths (not links) are not cancelled until the package execution of Shield.

HMI archive data

The following property is available:

Preselection offline archive	○ List selection ◉ During runtime	Under "Preselection offline archive", you have the following options: • List selection You can preselect the stored files as a list. Vie the button ▪, you open the "Preselection offline archive (HMI archive data)" dialog. Under the preselection you can: – Insert a file as link – Select a file from the list – Remove an entry from the list An area originating from the "input archive" is therefore replaced. • During runtime The offline archive is selected during package execution. Store an absolute or relative path. Relative paths start with "./" and are relative to the storage location of the package. Absolute or relative paths (not links) are not cancelled until the package execution of Shield.

NC archive data

The following property is available:

Preselection offline archive	○ List selection ● During runtime	Under "Preselection offline archive", you have the following options: • List selection You can preselect the stored files as a list. Using button , open dialog "Preselection offline archive (NC archive data)". Under the preselection you can: – Insert a file as link – Select a file from the list – Remove an entry from the list An area originating from the "input archive" is therefore replaced. • During runtime The offline archive is selected during package execution. Store an absolute or relative path. Relative paths start with "./" and are relative to the storage location of the package. Absolute or relative paths (not links) are not cancelled until the package execution of Shield.

PLC archive data

The following property is available:

Preselection offline archive	○ List selection ● During runtime	Under "Preselection offline archive", you have the following options: • List selection An archive can be preselected from which only the PLC area for package execution is used. Using button , open dialog "Preselection offline archive (PLC archive data)". Under the preselection you can: – Insert a file as link – Select a file from the list – Remove an entry from the list An area originating from the "input archive" is therefore replaced. • During runtime The offline archive is selected during package execution. Store an absolute or relative path. Relative paths start with "./" and are relative to the storage location of the package. Absolute or relative paths (not links) are not cancelled until the package execution of Shield.

DRV archive data/drive configuration

The following property is available:

Preselection offline archive	○ List selection ● During runtime	Under "Preselection offline archive", you have the following options: • List selection An archive can be preselected from which only the DRV area for package execution is used. Using button ▪, open dialog "Preselection offline archive (DRV archive data/drive configuration)". Under the preselection you can: – Insert a file as link – Select a file from the list – Remove an entry from the list An area originating from the "input archive" is therefore replaced. • During runtime The offline archive is selected during package execution. Store an absolute or relative path. Relative paths start with "./" and are relative to the storage location of the package. Absolute or relative paths (not links) are not cancelled until the package execution of Shield.

System settings

The following property is available:

Preselection offline archive	○ List selection ● During runtime	Under "Preselection offline archive", you have the following options: • List selection You can preselect an archive from which only the "Area of system settings" is used for the package execution. Using button ▪, open dialog "Preselection offline archive (system settings)". Under the preselection you can: – Insert a file as link – Select a file from the list – Remove an entry from the list An area originating from the "input archive" is therefore replaced. • During runtime The offline archive is selected during package execution. Store an absolute or relative path. Relative paths start with "./" and are relative to the storage location of the package. Absolute or relative paths (not links) are not cancelled until the package execution of Shield.

Output archive

The following property is available:

File path	You can specify a file path for the output archive comprising path and name. If only one folder path is specified, the package saves the output archive to this path as "Update.dsf". If nothing is entered in the Properties box, an "Update.dsf" output archive is created next to the package.

Descriptions of properties / buttons

Properties

The properties for the dialog can be found to the right of the "Project explorer" viewlet in the Home area. If you select a property, the description for this is displayed in the "Details" tab as a separate viewlet below the properties.

Buttons

There are buttons in the Home area below the properties:

- Use the left button to navigate back to the previous dialog
- Use the right button to navigate to the following dialog

8.3.11 Select offline archive (package execution)

This dialog specifies from where the output data for the NC, PLC and DRV control areas is obtained for the offline package execution without NCU.

Dialog view during package execution (for commissioning engineers / operators)

The "Select offline archive" dialog page cannot be deactivated. The dialog is **always** automatically executed when executing a package under Windows that contains the "Archive area" and the "PLC area".

More information on package execution can be found in the section under "Package execution (Page 401)".

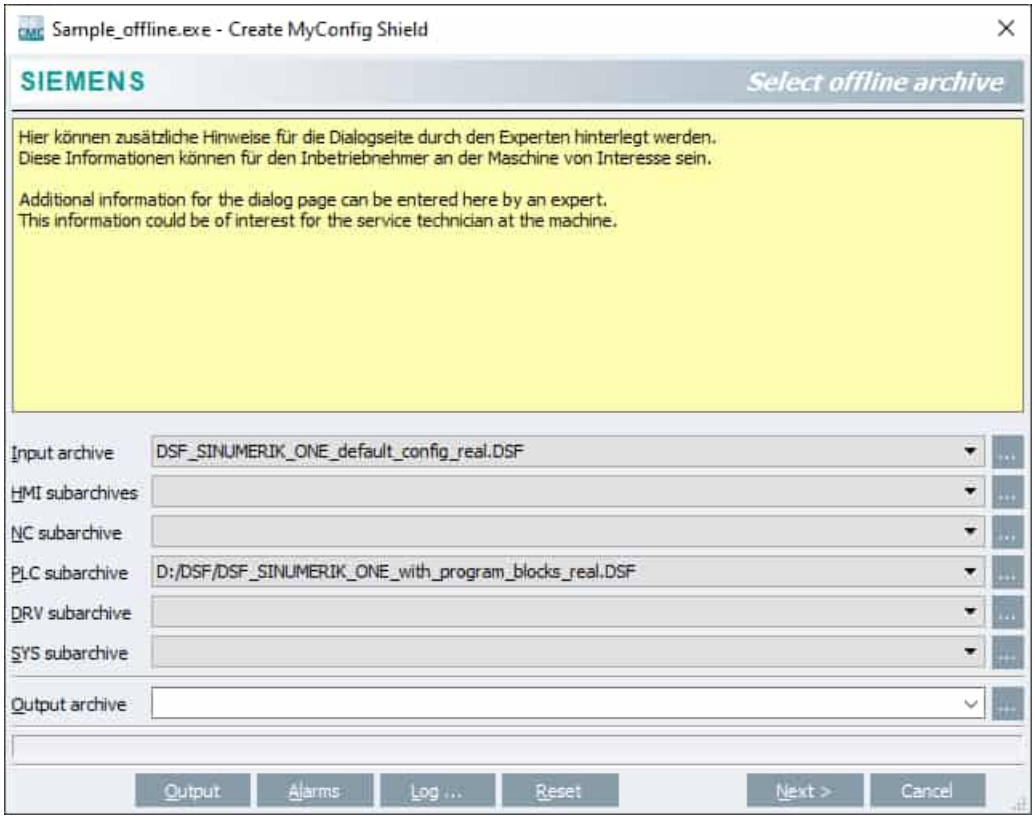


Figure 8-30 Select offline archive - Commissioning engineer / operator view

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons (Page 258)".

8.3.12 Backup initial state (configuration)

With this dialog, you perform a data backup of the initial state before the package execution.

You can back up all current control data, including the SD card **or** only the currently active online data of the NC, PLC and DRV subsystems, as SINUMERIK archive.

You configure the following property groups for the dialog page "Backup initial state":

- Basic dialog settings (Page 220)
- Dialog

- Backup SD card
- Archive

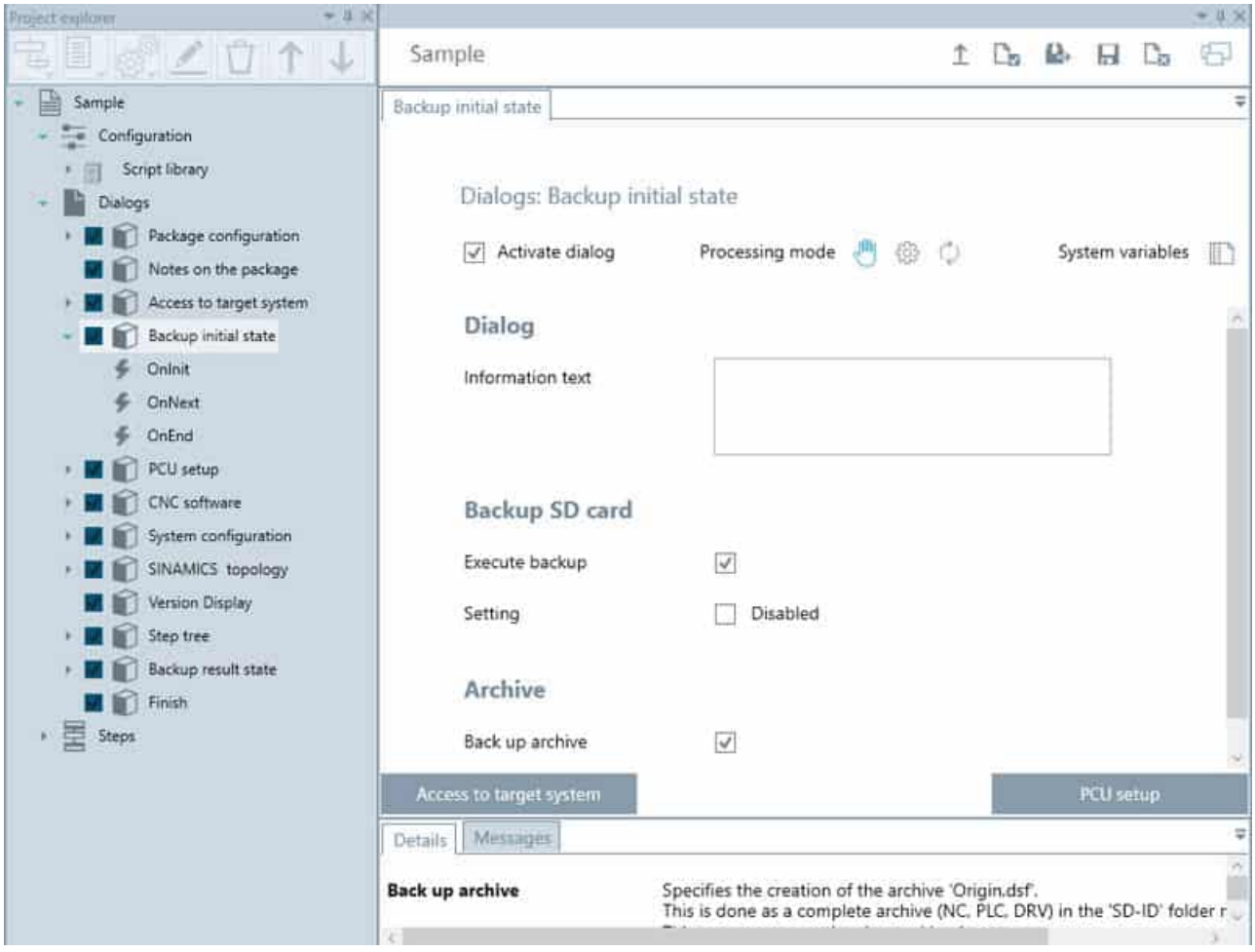


Figure 8-31 Backup initial state - project engineer view

Note

Specifying data areas

When selecting the "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined.

You can find an overview of this topic in the section "Dependencies of the dialog pages on data areas (Page 240)".

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the Backup initial state dialog (Page 385)".

Dialog

The following setting is available for "Dialog":

Information text	Insert a text for the operation in the field next to "Information text". Formatting with HTML is possible.

Backup SD card

The following settings are available for the "Backup SD card" property:

Execute backup	☑	You select a data backup of the NCU/PPU on this dialog page at the start of package execution. The content of the SD card, including the current active archive data (NC, PLC, DRV) is backed up in a folder named after the SD card ID. If a data backup is found next to the package when starting, you can select this data backup in the package selection for restoration. When the backup is restored, the backed-up archive data is reloaded automatically. In addition to the data backup "*.tgz", this folder also contains instructions "*.txt" and a package "*.usz" for a LINUX package or "*.exe" for a Windows package, for restoring the data backup. A confirmation prompt appears if a backup not belonging to this SD card ID is to be restored. You can find more information in the section "Package execution (Page 401)". Click the check box to activate the function. By default, the check box is deactivated.
Setting	☑ Disabled	The settings made during the configuration can only be viewed by the operator during execution, but not changed. Click the check box to activate the function. By default, the check box is deactivated.

Note

For "Setting locked" property

The setting for selecting the data backup can be disabled for the operating procedure.

The licenses of the control are also backed up. When restoring a backup, a query appears as to whether the licenses from the backup are to be restored, or the licenses on the control are to be retained.

Archive

Back up archive	☑	A complete archive can also be created for the diagnostics, with the setting that all data is to be backed up. The complete archive is stored in a folder whose name is based on the SD card ID with the name "Origin<Date>.dsf". Click the check box to activate the function. By default, the check box is deactivated.

Descriptions of properties / buttons

Properties

The properties for the dialog can be found to the right of the "Project explorer" viewlet in the Home area. If you select a property, the description for this is displayed in the "Details" tab as a separate viewlet below the properties.

Buttons

There are buttons in the Home area below the properties:

- Use the left button to navigate to the previous dialog
- Use the right button to navigate to the following dialog

8.3.13 Backup initial state (package execution)

This dialog allows a data backup of the NCU/PPU initial state to be made prior to package execution.

You can save all current control data, including the SD card or only the currently active online data of the subsystems (NC,PLC, DRV, HMI, SYS) as SINUMERIK archive.

Dialog view during package execution (for commissioning engineers / operators)

On this dialog page you can select a data backup of the SD card, including all current control data (NC, PLC, DRV, HMI and SYS). After package execution, the data backup is saved with the name '<SD card ID>_origin.tgz' in the folder with the SD card ID name next to the package.

More information on package execution can be found in the section under "Package execution (Page 401)".

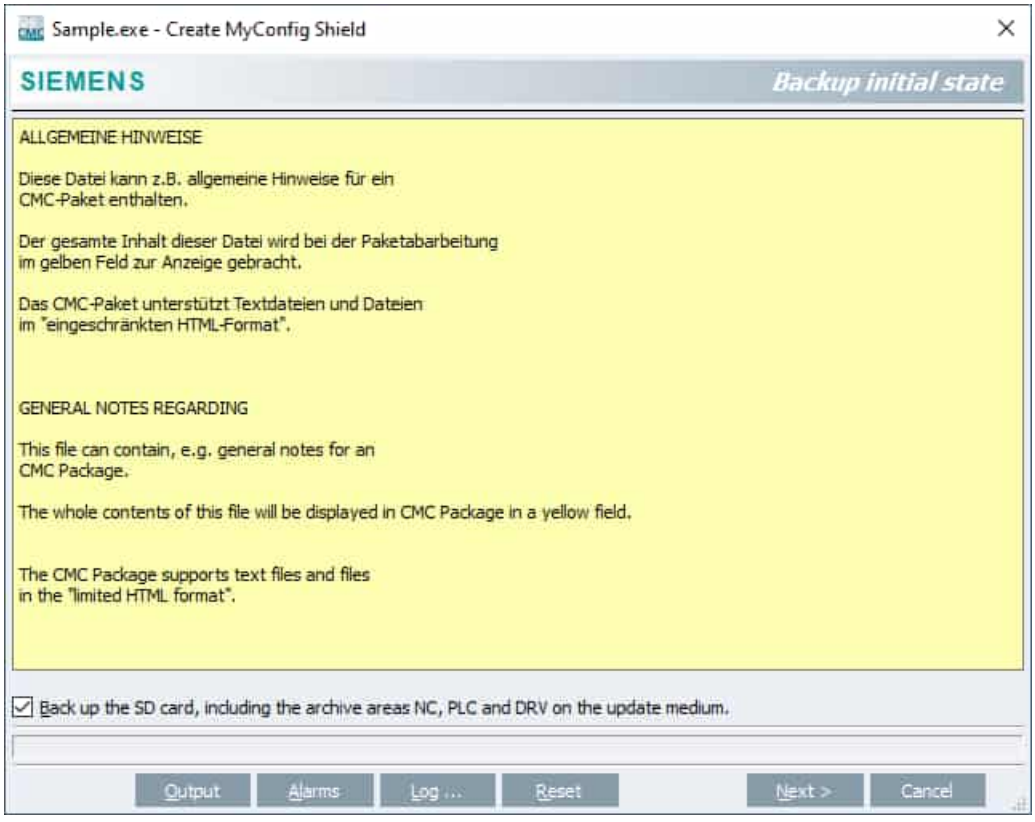


Figure 8-32 Backup initial state - commissioning engineer/operator view

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons (Page 258)".

8.3.14 PCU setup (configuring)

In the "PCU setup" dialog, you can create scripts to events or link existing scripts. In "PCU setup", you create actions of the "Execute" type.

You can create main steps with only one action of the type "Execute" at a time. The "Execute" action can be created under a step or an already existing action can be linked.

You can only create main steps in the form of a step list, substeps are not possible.

You can configure the following property groups for the "CNC software" dialog page:

- Basic dialog settings (Page 220)

No other properties can be configured on the dialog page.

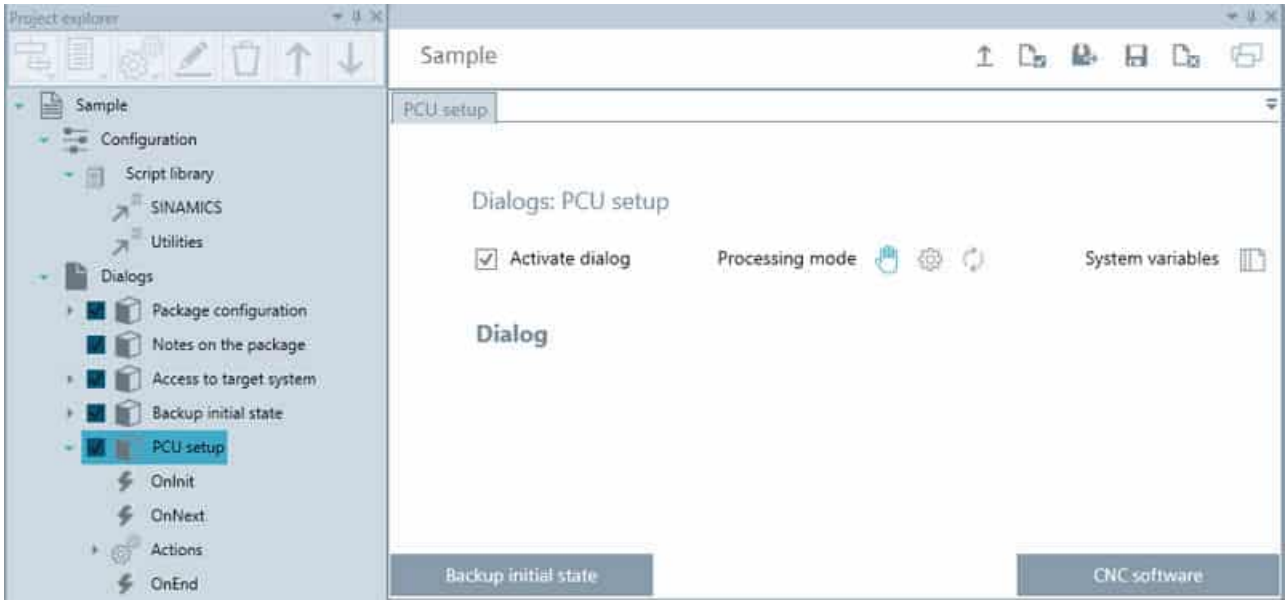


Figure 8-33 PCU setup - project engineer view

Note

Specifying data areas

When selecting the "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined in the "Dialog" tab.

You can find an overview of this topic in the section "Dependencies of the dialog pages on data areas".

8.3.15 PCU setup (package execution)

This dialog specifies with which PCU and which login data the package should be connected for the execution.

Dialog view during the package execution (for commissioning engineers / operators)

If you activated the "Use PCU" area in the "Configuration (Page 236)" under "Data areas", this dialog page is automatically processed when a package is being executed.

More information on package execution can be found in Section "Package execution (Page 401)".

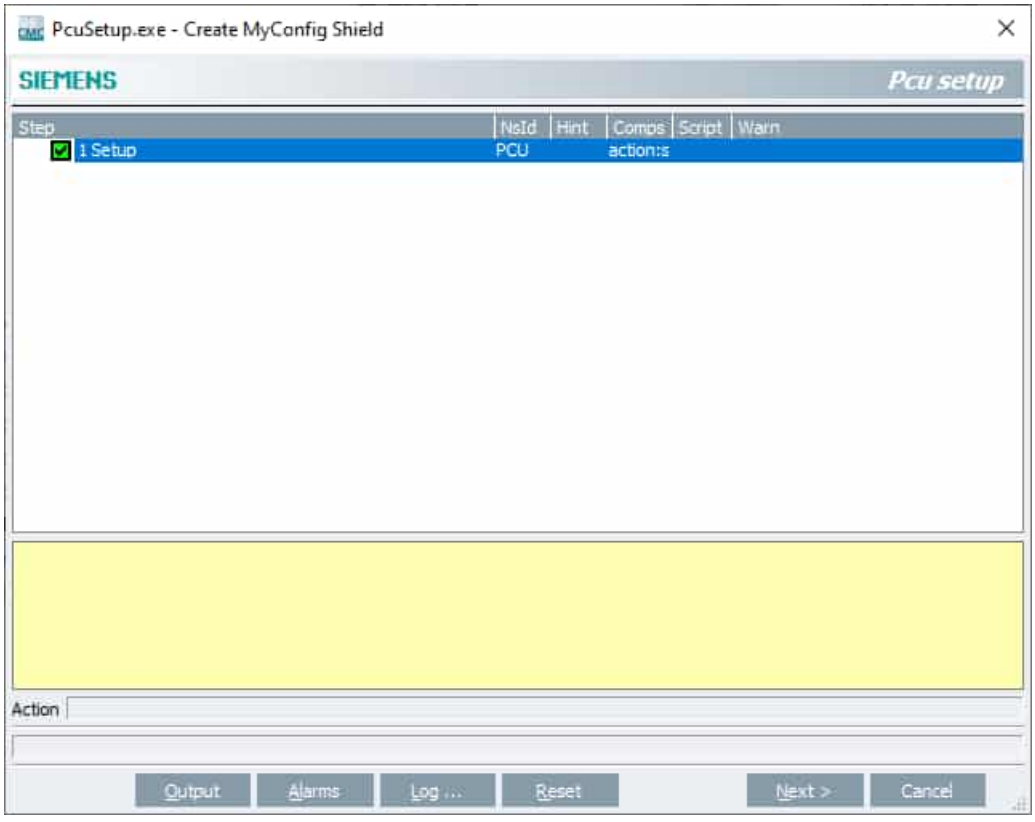


Figure 8-34 PCU setup - commissioning engineer / operator view

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons".

8.3.16 CNC software (configuration)

You use dialog "CNC software" to install the CNC software on the NCU.

You can configure the following property groups for the "CNC software" dialog page:

- Basic dialog settings (Page 220)
- Dialog
- Installation

- CNC software (*.tgz)
- Additional software (*.tgz)

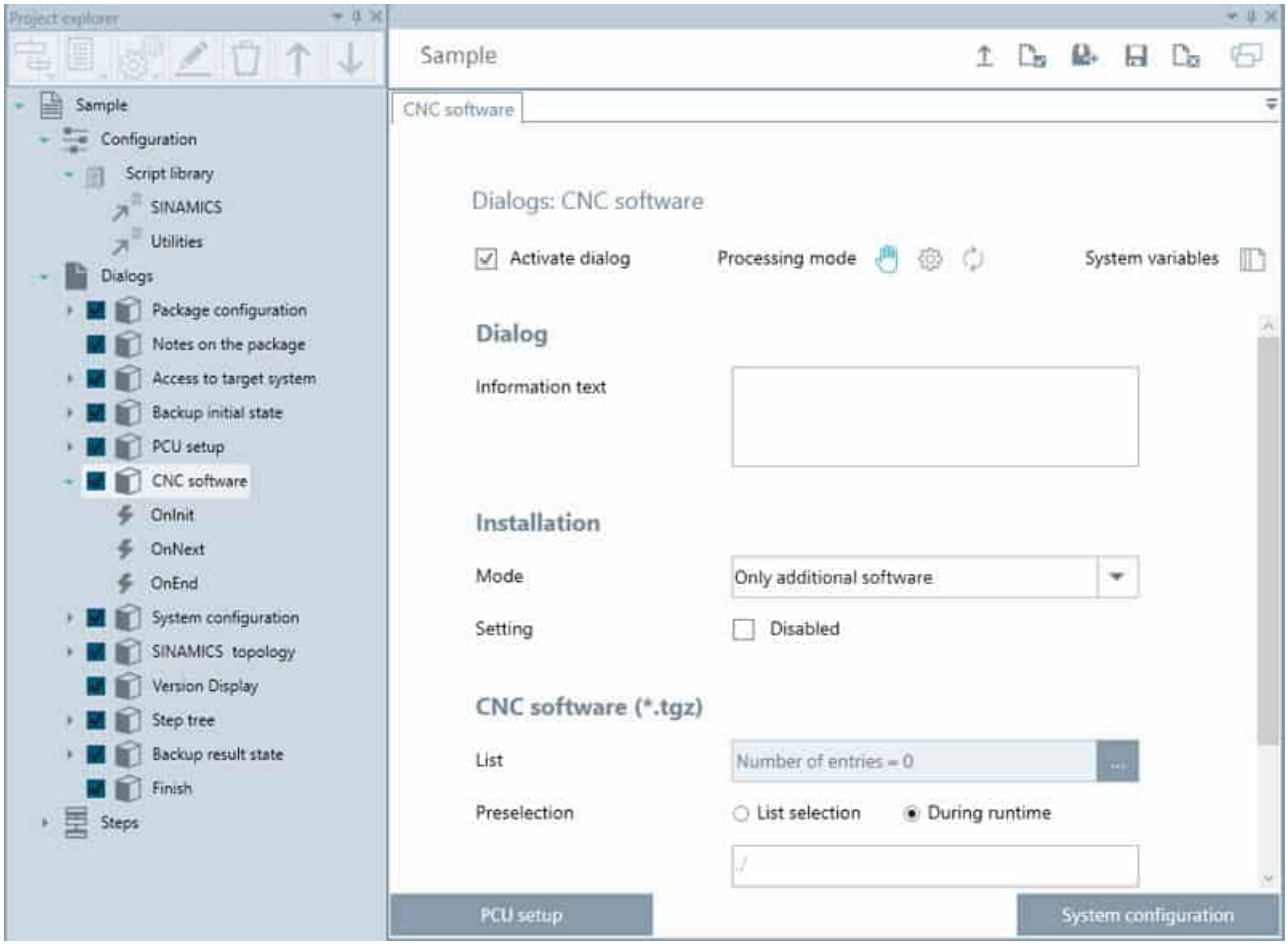


Figure 8-35 CNC software - project engineer view

Note

Specifying data areas

When selecting the "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined in the "Dialog" tab.

You can find an overview of this topic in the section "Dependencies of the dialog pages on data areas (Page 240)".

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the CNC software dialog (Page 386)".

Dialog

The following setting is available for "Dialog":

Table 8-15 "Dialogs" > Property group "Dialog"

Information text	Insert a text for the operation in the field next to "Information text". Formatting with HTML is possible.

Installation

The following settings are available for the "Installation" property:

Mode		Select the type of the installation.
	None	The software is not installed.
	New installation	The SD card is deleted and the system software is reinstalled. An overall reset is performed for the NC subsystems and the default data is loaded. **Note** Data migration means that the backed-up archive is imported from the previous firmware version to the newly installed firmware version. The step execution is performed later based on the archive data of the newly installed firmware version.
	Upgrade	The CNC software is replaced by the specified SINUMERIK SW version. During an upgrade user data is retained.
	Only additional software	Only the listed additional "*.tgz" packages without CNC software installation or upgrade are installed. Only elements from the list of "Additional software (*.tgz)" are taken into account.
Setting	☐ Locked	For the operating workflow, you can lock the setting for the "mode". Click the check box to activate the function. By default, the check box is deactivated.

CNC software (*.tgz)

The following settings are available for the "CNC software (*.tgz)" property:

List		TGZ files that contain the CNC software are provided by SIEMENS Support. You can insert the TGZ files in a list as link. Linked files are not taken into this list until the package is deployed.
Preselection	List selection During runtime	Under "Preselection", you have the following options: • List selection You can preselect the stored files as a list. You open the "Preselection (CNC software (*.tgz))" dialog via the button . Under "Preselection" you can: – Insert a file as link – Select a file from the list – Remove an entry from the list – Move a file entry in the list either up or down • During runtime The CNC software is selected during package execution. Store an absolute or relative path. Relative paths start with "./" and are relative to the storage location of the package. Absolute or relative paths (not links) are not cancelled until the package execution of Shield.

Note

If you start the package from a service system USB flash drive, then generally booting is from the SD card after this dialog page.

Additional software (*.tgz)

Preferably, this function is used when installing SINAMICS TEC. You can also install any TGZ files.

The following settings are available for the "Additional software (*.tgz)" property:

List	To install several TGZ files in a sequence, you must save the required "*.tgz" files in this list. The TGZ files are then automatically installed in the list from the top to the bottom in the sequence that you defined. You define the sequence of the TGZ files to be installed by selecting a file and then navigating to the required position using the "Move up" or "Move down" buttons. You insert the TGZ files in the list as link. The linked TGZ files are transferred into the package when the package is deployed.

Descriptions of properties / buttons

Properties

The properties for the dialog can be found to the right of the "Project explorer" viewlet in the Home area. If you select a property, the description for this is displayed in the "Details" tab as a separate viewlet below the properties.

Buttons

There are buttons in the "Home area" below the properties:

- Use the left button to navigate to the previous dialog
- Use the right button to navigate to the following dialog

8.3.17 CNC software (package execution)

You use dialog "CNC software" to newly install or upgrade the CNC software on the NCU/PPU. You can also load using the "Only additional software" option.

Dialog view during the package execution (for commissioning engineers / operators)

The "CNC software" dialog installs new CNC software or upgrades existing CNC software.

More information on package execution can be found in the section under "Package execution (Page 401)".

Note

Downgrade of the CNC system software

A downgrade of the CNC system software might not be possible. The restore function is not affected by this.

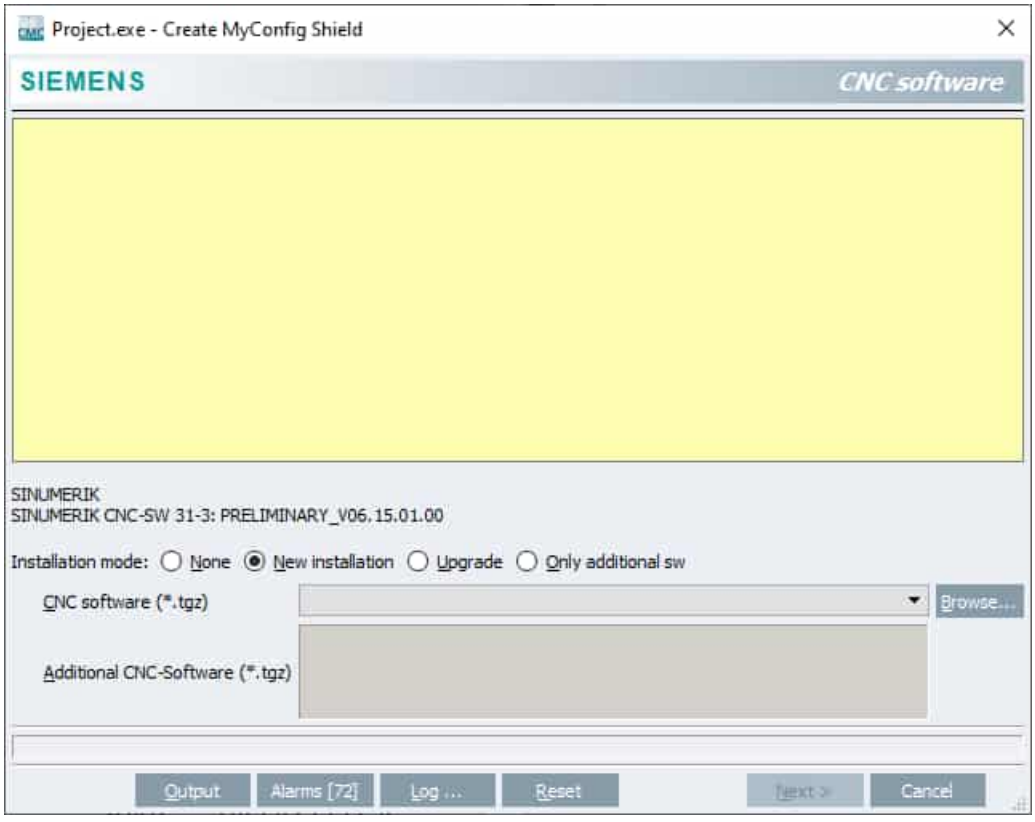


Figure 8-36 CNC software - commissioning engineer / operator view

Via the "CNC software (*.tgz)" drop-down list you select a file that was assigned to the package or a file located next to the package (prefix: "./").

The "Browse" button allows a selection to be made from any location.

When you select the option "Only additional software", configured files are displayed in the field "Additional CNC software (*.tgz)".

The files are configured on the "CNC software (configuring)" dialog page in the "Additional software (*.tgz)" property.

After the TGZ file has been imported, the NCU/PPU is restarted automatically.

Note

TGZ files generated using "sc backup xyz.tgz" or backups generated using Create MyConfig can **only** be used for a new installation.

If a backup of CMC is used for a new installation, then the archive data it contains is imported.

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons (Page 258)".

8.3.18 System configuration (configuration)

In this dialog, specify which data of the NC, PLC (HW) and DRV subsystems is used as output data for the package execution.

With regard to drive commissioning, this dialog corresponds to the "SINAMICS topology" dialog.

You can configure the following property groups for the "System configuration" dialog page:

- Basic dialog settings (Page 220)
- Dialog
- Offline archives
- NC archive data
- PLC archive data
- DRV archive data/drive configuration
- HMI archive data
- System settings

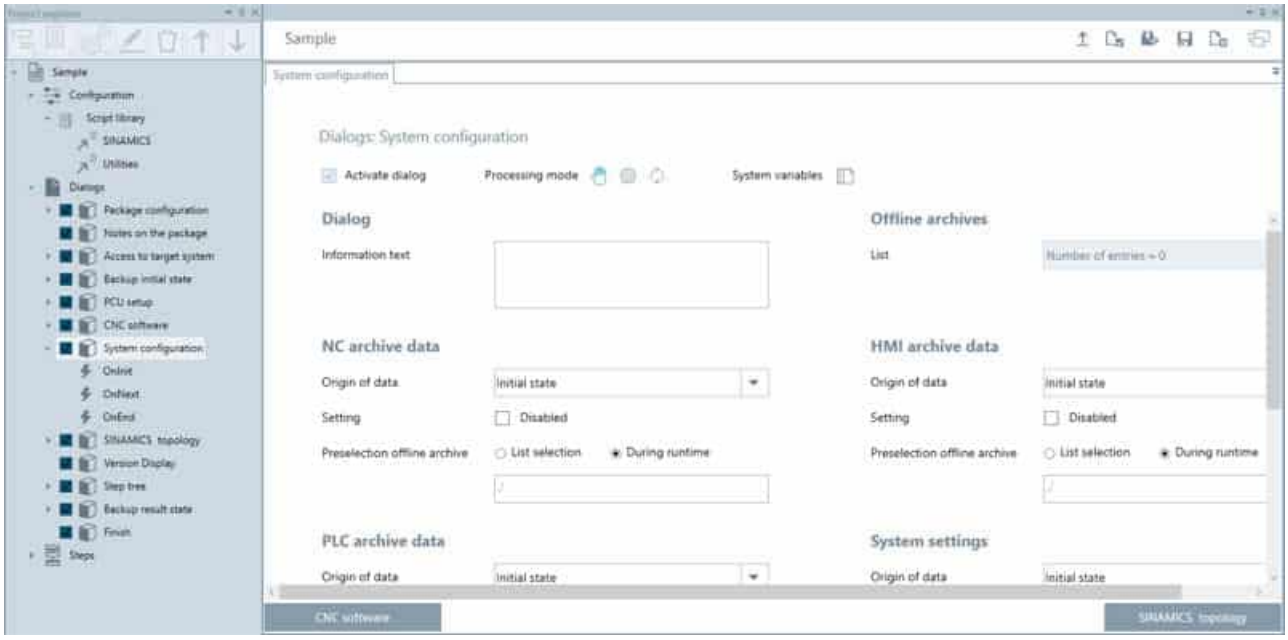


Figure 8-37 System configuration

Note

Specifying data areas

When selecting the "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined.

You can find an overview of this topic in the section "Dependencies of the dialog pages on data areas (Page 240)".

Overview of the "System configuration" dialog

The source and/or output data is made available via dialog page "System configuration".

You subsequently manipulate the data in dialog pages "SINAMICS topology (Page 299)" and "Step tree (Page 312)".

Once the "Step tree" dialog page has been executed, the modified data is transferred to the control or into a result archive.

In the "System configuration" dialog page, you can define for every archive area as to whether this is used.

For "Origin of data", select from where the data of the individual areas should be sourced.

The "Origin of data" can be blocked individually for each area by the project engineer in order to prevent changes by the operator during the package execution.

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the System configuration dialog (Page 387)".

Dialog

The following setting is available for "Dialog":

Information text	Insert a text for the operation in the field next to "Information text". Formatting with HTML is possible.

Offline archives

The following property is available for the "offline archives":

List		The list of all "offline archives" to be included in the package is displayed in dialog "List (offline archives)". Open the list via the button.
		The adjacent toolbar is available in dialog "List (offline archives)": • You can link SINUMERIK archives with NC data. These archives are also saved in the package during deployment and can be selected during package execution. • You can link several archives, which are subsequently listed for preselection and can be selected during package execution.
		Insert as link
		Remove the selected entry from the list
		Move the selected entry up or down
		

NC archive data

The following settings are available for "NC archive data". The settings become effective when the package is being executed.

Table 8-16 NC archive data

Origin of data	**Note** This selection can be blocked by the project engineer in order to prevent changes by the operator during package execution.	
	Initial state	If, in the "Project explorer" viewlet under "Configuration (Page 238)", you activate the property "Data areas" > "Use NCU/PPU", then the archive data can be directly loaded from the control with the "Initial state" setting. If the initial state is used, the archive is extracted that was located on the control when this dialog was first entered. If the CNC software was installed beforehand, only default data is available in the archive. **Note** Data is not reset when upgrading the CNC software. The archive is used that is available in the system after data migration.
	Memory reset	With this option, default data is loaded and then the archive data is backed up.
	Offline archive	With this option, the output data for the NC area is taken from a stored SINUMERIK archive.
	Do not use	With this option, no output data for this area is made available by the dialog page. Data of the area is also not transferred to the control. More information can be found in the note under the section "Archive data transfer (package execution) (Page 316)".
Setting	☐ Locked	In order to prevent changes by the operator during package execution, the project engineer can block the selection for "Origin of data" under "Setting". To block the selection for the operator, activate the check box. By default, the check box is deactivated.

Preselection offline archive	○ List selection ● During runtime	Choose between the options: • List selection If an archive is preselected, this is included in the package. During the check run, a check is made as to whether the archive is available. • During runtime "During runtime" is activated by default. If the "During runtime" option is enabled, enter an archive path. The archive is not part of the package and must only be available at runtime. You can store an absolute or relative path. Relative paths start with "./" and refer to the storage location of the package. Absolute or relative paths are not cancelled until the package execution of Shield. If the path is specified during runtime, the path is not checked for validity by the check run.
		Under List selection, preselect an archive from which only the NC area is used for package execution. In the "Preselection offline archive (NC archive data)" dialog, you can: • Select an DSF archive from the list • Insert a DSF archive as a link **Note** If data for "Offline archive" must be entered in the step tree via components, no SINUMERIK archive must be specified in the preselection.

PLC archive data

The following settings are available for the "PLC archive data" and are activated when the package is executed:

Origin of data	**Note** This selection can be blocked by the project engineer in order to prevent changes by the operator during package execution.	
	Initial state	If, in the "Project explorer" viewlet under Configuration (Page 238), you activate the property "Data areas" > "Use NCU/PPU", then the archive data can be directly loaded from the control with the "Initial state" setting. **Note** Data is not reset when upgrading the CNC software. The archive that is available in the system after data migration is used.
	Offline archive	With this option, the output data for the PLC area is taken from a stored SINUMERIK archive.
	Do not use	With this option, no output data for this area is made available by the dialog page. Data of the area is also not transferred to the control. More information can be found in the note under the section "Archive data transfer (package execution) (Page 316)".
Setting	☐ Locked	In order to prevent changes by the operator during package execution, the project engineer can block the selection for "Origin of data" under "Setting". To block the selection for the operator, activate the check box. By default, the check box is deactivated.

Preselection offline archive	○ List selection ● During runtime	Choose between the options: • List selection If an archive is preselected, this is included in the package. During the check run, a check is made as to whether the archive is available. • During runtime "During runtime" is activated by default. If the "During runtime" option is enabled, enter an archive path. The archive is not part of the package and must only be available at runtime. You can store an absolute or relative path. Relative paths start with "./" and refer to the storage location of the package. Absolute or relative paths are not cancelled until the package execution of Shield. If the path is specified during runtime, the path is not checked for validity by the check run.
		Under List selection, preselect an archive from which only the PLC area is used for package execution. In the "Preselection offline archive (PLC archive data)" dialog, you can: • Select an DSF archive from the list • Insert a DSF archive as a link **Note** When data for "Offline archive" is entered in the step tree via components, an archive from the overall reset state must be saved here or the "Origin of data" must be set to "Memory reset".

DRV archive data/drive configuration

In the "DRV archive data/drive configuration" property group, you can specify via the "Origin of data" property whether drive commissioning will be performed in the production step by "Automatic configuration" or "User-specified topology".

The following settings are available to you for the "DRV archive data/drive configuration". The settings take effect when the package is executed.

Origin of data	**Note** This selection can be blocked by the project engineer in order to prevent changes by the operator during package execution.	
	Initial state	If, in the "Project explorer" viewlet under Configuration (Page 236), you activate the property "Data areas" > "Use NCU/PPU", then the archive data can be directly loaded from the control with the "Initial state" setting. **Note** Data is not reset when upgrading the CNC software. The archive is used that is available in the system after data migration. When reinstalling the CNC software, an empty DRV archive would result.
	Automatic configuration	The "Automatic configuration" selection runs through the drive unit configuration during the package execution in the same way as SINUMERIK Operate. "Automatic configuration" mode initially restores the SINAMICS factory settings. Then, automatic HMI device configuration (including DO sorting, BICO interconnection, etc.) takes place. In this mode, you can work with a "comparison topology (*.ust)", which must be pasted in the "SINAMICS topology" dialog page. This means that the automatically identified SINAMICS target topology can be compared with a comparison topology generated offline. During "Automatic configuration", SINAMICS determines the target topology that specifies all modules connected at this time, including their DRIVE-CLiQ wiring. In the subsequent package execution, the most recently created online archive is always used. It is only possible to change the topology by adding, removing or replacing components via SINUMERIK Operate. Or you must perform a configuration again. **Note** **First encoder on the CU or downstream of the HUB** If during a drive data generation via "Automatic configuration", a first encoder (SMI or SMC) is connected downstream of a HUB or to the CU, then a complete automatic configuration of the drive by SINAMICS and Shield is not possible. Parameter "P1982" (Pole position identification on) is set to "1" by SINAMICS instead of "0" and cannot be corrected automatically here. The user can correct the parameter manually or also in the step tree of a CMC package. Set parameter "P1982=0" in the step tree in a component for all drives whose first encoder is connected downstream of the HUB or to the CU.
	User-specified topology	With the selection "user-specified topology", a user-specified topology (*.utz2) is loaded during package execution. For drive commissioning, a user-specified topology can be loaded that may also contain modules that will not be added, wired and activated until later during commissioning. The necessary data objects are created by SINAMICS, also for the modules that are not yet present, so that the user data can already be loaded during package execution. In the subsequent package execution, the most recently created online archive is always used.

		A main application case for this are DQI/SMI motors and encoders. The Cu and the NX expansion modules must always be present. **Note** When selecting the "User-specified topology" option, you must also enter the appropriate user-specified topology (*.utz2) in the "SINAMICS topology" dialog.
	Offline archive	When the "Offline archive" is set, the data from a SINUMERIK archive is used.
	Do not use	With this option, no output data for this area is made available by the dialog page. Data of the area is also not transferred to the control. More information can be found in the note under the section "Archive data transfer (package execution) (Page 316)".
Setting	☐ Locked	In order to prevent changes by the operator during package execution, the project engineer can block the selection for "Origin of data" under "Setting". To block the selection for the operator, activate the check box. By default, the check box is deactivated.
Preselection offline archive	○ List selection ◉ During runtime	Choose between the option buttons: • List selection If an archive is preselected, this is included in the package. During the check run, a check is made as to whether the archive is available. • During runtime "During runtime" is activated by default. If the "During runtime" option is enabled, enter an archive path. The archive is not part of the package and must only be available at runtime. You can store an absolute or relative path. Relative paths start with "./" and refer to the storage location of the package. Absolute or relative paths are not cancelled until the package execution of Shield. If the path is specified during runtime, the path is not checked for validity by the check run.
		Under List selection, preselect an archive from which only the DRV area is used for package execution. Under "Preselection offline archive (DRV archive data/drive configuration)", you can: • Select an DSF archive from the list • Insert a DSF archive as a link

HMI archive data

The following settings are available for "HMI archive data", and are activated when the package is executed:

Table 8-17 HMI archive data

Origin of data	**Note** This selection can be blocked by the project engineer in order to prevent changes by the operator during package execution.	
	Initial state	If, in the "Project explorer" viewlet under "Configuration (Page 238)", you activate the property "Data areas" > "Use NCU/PPU", then the archive data can be directly loaded from the control with the "Initial state" setting. If the initial state is used, the archive is extracted that was located on the control when this dialog was first entered. If the CNC software was installed beforehand, only default data is available in the archive.
	Offline archive	When using this option, the output data for area "HMI" is taken from a saved SINUMERIK archive.
	Do not use	With this option, no output data for this area is made available by the dialog page. Data of the area is also not transferred to the control. More information can be found in the note under the section "Archive data transfer (package execution) (Page 316)".
Setting	☐ Locked	In order to prevent changes by the operator during package execution, the project engineer can block the selection for "Origin of data" under "Setting". To block the selection for the operator, activate the check box. By default, the check box is deactivated.

Preselection offline archive	○ List selection ● During runtime	Choose between the options: • List selection If an archive is preselected, this is included in the package. During the check run, a check is made as to whether the archive is available. • During runtime "During runtime" is activated by default. If the "During runtime" option is enabled, enter an archive path. The archive is not part of the package and must only be available at runtime. You can store an absolute or relative path. Relative paths start with "./" and refer to the storage location of the package. Absolute or relative paths are not cancelled until the package execution of Shield. If the path is specified during runtime, the path is not checked for validity by the check run.
		Under List selection, preselect an archive from which only the HMI area is used for package execution. In the "Preselection offline archive (HMI archive data)" dialog, you can: • Insert a DSF archive as a link • Select an DSF archive from the list

System settings

The following settings are available for the "System settings". The settings take effect when the package is executed.

Origin of data	**Note** This selection can be blocked by the project engineer in order to prevent changes by the operator during package execution.	
	Initial state	If, in the "Project explorer" viewlet under "Configuration (Page 236)", you activate the property "Data areas" > "Use NCU/PPU", then the archive data can be directly loaded from the control with the "Initial state" setting. **Note** Data is not reset when upgrading the CNC software. The archive is used that is available in the system after data migration. When reinstalling the CNC software, an empty DRV archive would result.
	Offline archive	When the "Offline archive" is set, the data from a SINUMERIK archive is used.
	Do not use	With this option, no output data for this area is made available by the dialog page. Data of the area is also not transferred to the control. More information can be found in the note under the section "Archive data transfer (package execution) (Page 316)".

Setting	☐ Locked	In order to prevent changes by the operator during package execution, the project engineer can block the selection for "Origin of data" under "Setting". To block the selection for the operator, activate the check box. By default, the check box is deactivated.
Preselection offline archive	○ List selection ◉ During runtime	Choose between the option buttons: • List selection If an archive is preselected, this is included in the package. During the check run, a check is made as to whether the archive is available. • During runtime "During runtime" is activated by default. If the "During runtime" option is enabled, enter an archive path. The archive is not part of the package and must only be available at runtime. You can store an absolute or relative path. Relative paths start with "./" and refer to the storage location of the package. Absolute or relative paths are not cancelled until the package execution of Shield. If the path is specified during runtime, the path is not checked for validity by the check run.
		Under List selection, preselect an archive from which only the system settings are used for package execution. Under "Preselection of the offline archive", you can: • Select a file from the list • Insert a file as link You can preselect archives that are only added to the package during runtime. These archives are not checked by the check run.

Descriptions of properties and buttons

Properties

The properties for the dialog can be found to the right of the "Project explorer" viewlet in the Home area viewlet. If you select a property, the description for this is displayed in the "Details" tab as a separate viewlet below the properties.

Buttons

There are buttons in the Home area below the properties:

- Use the left button to navigate to the previous dialog
- Use the right button to navigate to the following dialog

8.3.19 System configuration (package execution)

In this dialog, specify which data of the NC, PLC (HW) and DRV subsystems is used as output data for the package execution.

This dialog corresponds with regard to the drive commissioning to the "SINAMICS topology" dialog.

If the "System configuration" dialog page is executed in Edit mode "Manual", you can adapt the "Origin of data" for every area if the project engineer has not blocked this.

For "Offline archive", select the archives in the following ways:

- The archives are transferred with the package
- The archives are located next to the package (prefix "./")
- The archives are inserted from an arbitrary location via the button ("Browse").

Note

From package execution back to the system configuration

If, during package execution, a jump is made to the "System configuration" dialog page via "Back":

- The archives backed up in the system configuration are used again
- Changes that have possibly already been made are rejected.

Example

Example archive next to the package:

```
./myPath/myArchive.dsf
```

A "myArchive.dsf" archive from the "myPath" folder is used. It is located next to the package.

The "./" prefix represents the current path in which the package file is located at the time of execution.

Dialog view during package execution (for commissioning engineers / operators)

In this dialog you set which archive data (NC, PLC and DRV) will be used later in the subsequent package execution.

Note

If the "Archive area" is activated on the "Package" tab, the "System configuration" dialog must be parameterized.

To establish communication between the NCK, the integrated CU and the NX expansion modules, you must import an appropriate PLC hardware configuration in the form of an SDB archive.

More information on package execution can be found in the section of the same name "Package execution (Page 401)".

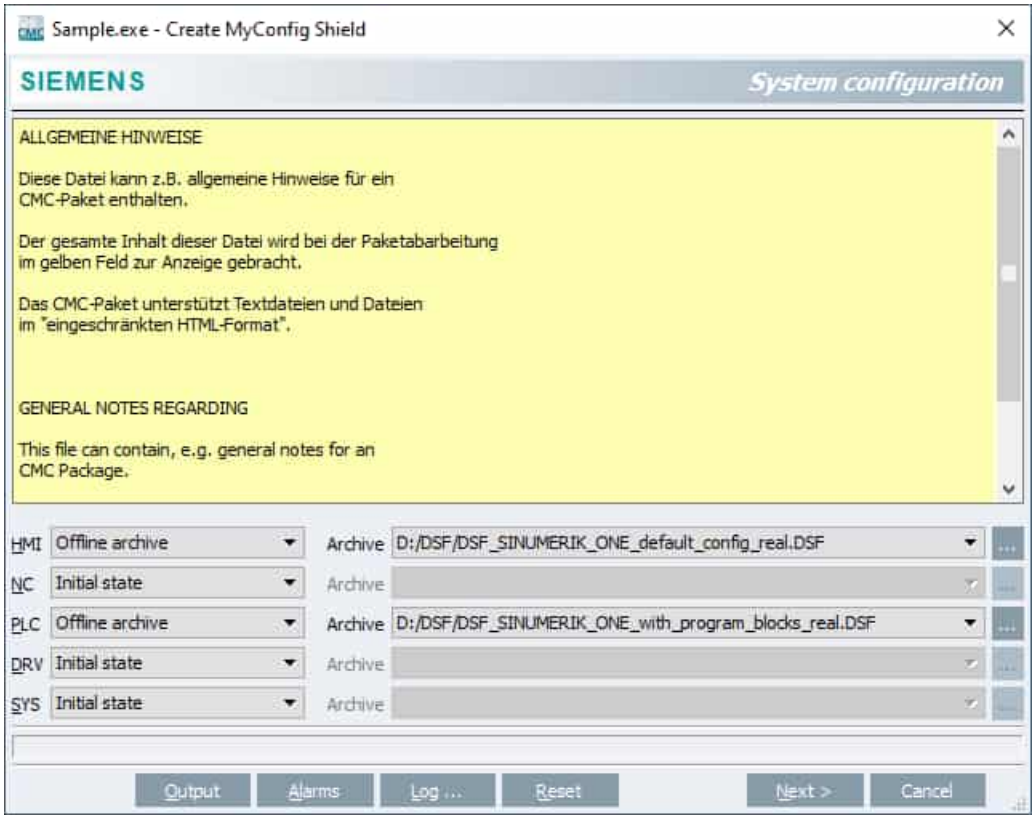


Figure 8-38 System configuration - commissioning engineer/operator view

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons (Page 258)".

8.3.20 SINAMICS topology (configuration)

In the "SINAMICS topology" dialog, you specify how and which user data must be provided for the SINAMICS topology in the control.

In the dialog, the "actual topology" can be compared with the "target topology". You can already create drive objects for modules that are not yet available.

This dialog corresponds to the "System configuration" dialog.

You can configure the following property groups for the "SINAMICS topology" dialog page:

- Basic dialog settings (Page 220)
- Axis-drive assignment
- Comparable topology (*.ust)
- User-specified topology (*.utz2)

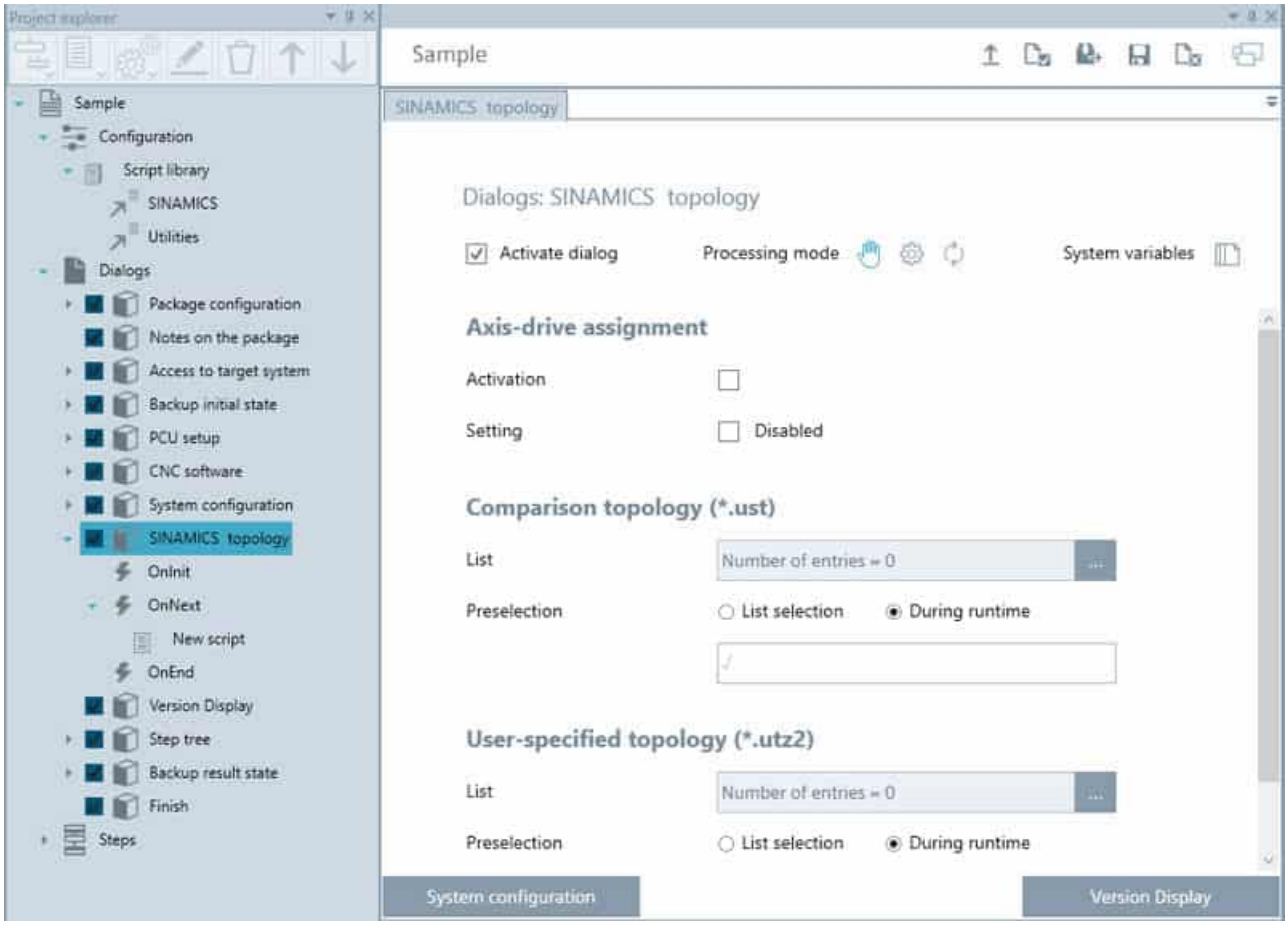


Figure 8-39 SINAMICS topology

Note

Specifying data areas

When selecting the "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined.

You can find an overview of this topic in Section "Dependencies of the dialog pages on data areas (Page 240)".

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the SINAMICS topology dialog (Page 387)".

Axis-drive assignment

The following properties available for the "axis-drive assignment" are activated during package execution:

Activation	☐	Specify whether you want to activate the automatic "axis-drive assignment". By default, the check box is deactivated.
Setting	☐ Disabled	By activating the check box, the configuration engineer specifies that the operator cannot make any changes during package execution. By default, the check box is deactivated.

An automatic Axis-drive assignment is only realized in variants "Comparable topology" and "User-specified topology", as only these variants use the appropriate configuration from Topo.

As a prerequisite, this Axis-drive assignment must have been configured and activated in Topo for the comparison or User-specified topology.

By activation, with equal rights, of the Axis-drive assignment in Topo, there is a guarantee that not only the expert for Expert, but also the expert for Topo will agree to this assignment.

Moreover, it is possible for the project engineer for Expert and the operator during package execution to deactivate the automatic Axis-drive assignment without having to open and edit the topology files.

Comparable topology (*.ust)

Note

Prerequisite for comparable topology

On dialog page "SINAMICS topology", a comparable topology is always used if, on the dialog page "System configuration" in the property group "DRV archive data/drive configuration" for "Origin of data", you selected one of the following properties:

- Initial state
- Automatic configuration
- Offline archive

With the comparable topology, the DRIVE-CLiQ wiring and components are checked based on the "SINAMICS target topology" from the "DRV archive".

The following properties available for the "comparable topology (*.ust)" are activated during package execution:

List		Using button , dialog "List (comparable topology (*.ust))". opens Select the desired topology from the list. You can insert comparable topologies as link under "List". These are saved in the package during deployment and can be selected during package execution.
Preselection	○ List selection ● During runtime	Under "Preselection", you have the following options: • List selection Select a Comparable topology (*.ust) that is to be used during package execution. Click the button to open the "List (comparable topology (*.ust))" dialog. Under Preselection you can: – Insert a file as link – Select a file from the list – Remove an entry from the list • During runtime The comparable topology is selected during package execution. Store an absolute or relative path. Relative paths start with "./" and are relative to the storage location of the package. Absolute or relative paths (not links) are not cancelled until the package execution of Shield.

User-specified topology (*.utz2)

Note

Prerequisite for user-specified topology

If, on the "System configuration" dialog page, user-specified topology has been selected at "DRV archive data/drive configuration" > "Origin of data", a configured user-specified topology always applies.

If no critical inconsistency with the real topology is found by the drive in the user-specified topology, the user-specified topology is applied as the target topology of SINAMICS.

The following properties are available for the "user-specified topology (*.utz2)" are activated during package execution:

List		Using button , dialog "List (user-specified topology (*.utz2))" opens. Select the required User-specified topology (*.utz2) from the list. You can insert user-specified topologies as link under "List". These are also saved in the package during deployment and can be selected during package execution.
Preselection	○ List selection ◉ During runtime	Under "Preselection", you have the following options: • List selection Select a user-specified topology (*.utz2) that is to be used for the package execution. Clicking the button opens the "List (user-specified topology (*.utz2))". Under Preselection you can: – Insert a file as link – Select a file from the list – Remove an entry from the list • During runtime The user-specified topology is selected during package execution. Store an absolute or relative path. Relative paths start with "./" and are relative to the storage location of the package. Absolute or relative paths (not links) are not cancelled until the package execution of Shield.

Note

The "SINAMICS topology" dialog differs during package execution depending on whether the drive configuration "User-specified topology" or one of the three other variants were selected in the "System configuration" dialog.

Descriptions of properties / buttons

Properties

The properties for the dialog can be found to the right of the "Project explorer" viewlet in the Home area. If you select a property, the description for this is displayed in the "Details" tab as a separate viewlet below the properties.

Buttons

There are buttons in the Home area below the properties:

- Use the left button to navigate to the previous dialog
- Use the right button to navigate to the following dialog

8.3.21 SINAMICS topology (package execution)

The "SINAMICS topology" dialog specifies how and which user data are required for the SINAMICS topology in the control.

The dialog checks the "actual topology" against the "target topology". Drive objects for modules that are not yet available can already be created here.

This dialog corresponds to the "System configuration" dialog.

Dialog view during package execution (for commissioning engineers / operators)

The 3 possible dialog versions are described in the following:

- Comparison topology
- User-specified topology
- Manual DO variable assignment

Comparison topology

The dialog version "Comparison topology" is used if one or all of the following options have been configured for the drives during device configuration (configuration):

- Initial state
- Offline archive
- Automatic configuration

The comparison topology must be selected in the "SINAMICS topology" dialog page.

The following diagram is an example of the dialog page using a comparison topology after automatic device configuration.

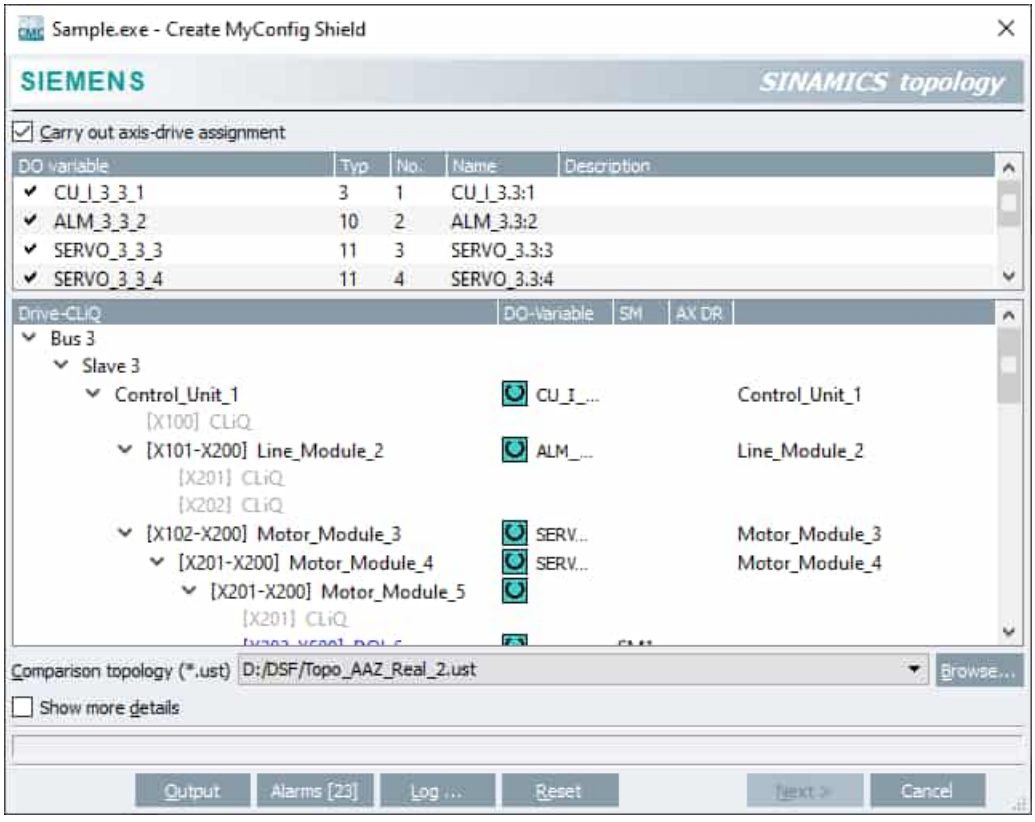


Figure 8-40 SINAMICS topology - commissioning engineer / operator view - example of comparison topology

Carrying out axis-drive assignment

The automatic axis-drive assignment is activated by clicking on the check box and thus placing a checkmark.

As a prerequisite, this Axis-drive assignment must have been configured and activated in Topo.

DO variables

The upper part of the dialog page provides a list of all DO variables that are in the DO variable list in the comparison topology.

Properties of the DO variables are listed in the list. A checkmark at the beginning of the line indicates whether a DO variable is used and was thus assigned to a drive object.

DRIVE-CLiQ wiring

The "DRIVE-CLiQ wiring" of the components and the assignments of the DO variables to the components are displayed in the center of the dialog page.

Column "SM" indicates whether the "Sensor Module" is the 1st or 2nd encoder. Sensor Modules are shown in blue.

Column "AX DR" includes information on the axis-drive assignment.

Comparison topology (*.ust)

Via the "Comparison topology (*.ust)" drop-down list, a file that was assigned to the package or a file located next to the package (prefix: "./") can be used.

A file from a different location and with another extension can be selected by you using the "Browse..." button, e.g. xml, from the E-CAD system.

Showing more details

In the comparison topology, you can hide or show more information for diagnostics in the list and tree view via the "Show more details" checkbox.

Status bar

The article number and the serial number of the selected component are displayed in the status bar above the buttons.

User-specified topology

This dialog version is used for the SINAMICS topology if "User-specified topology" was selected for the device configuration of the drives.

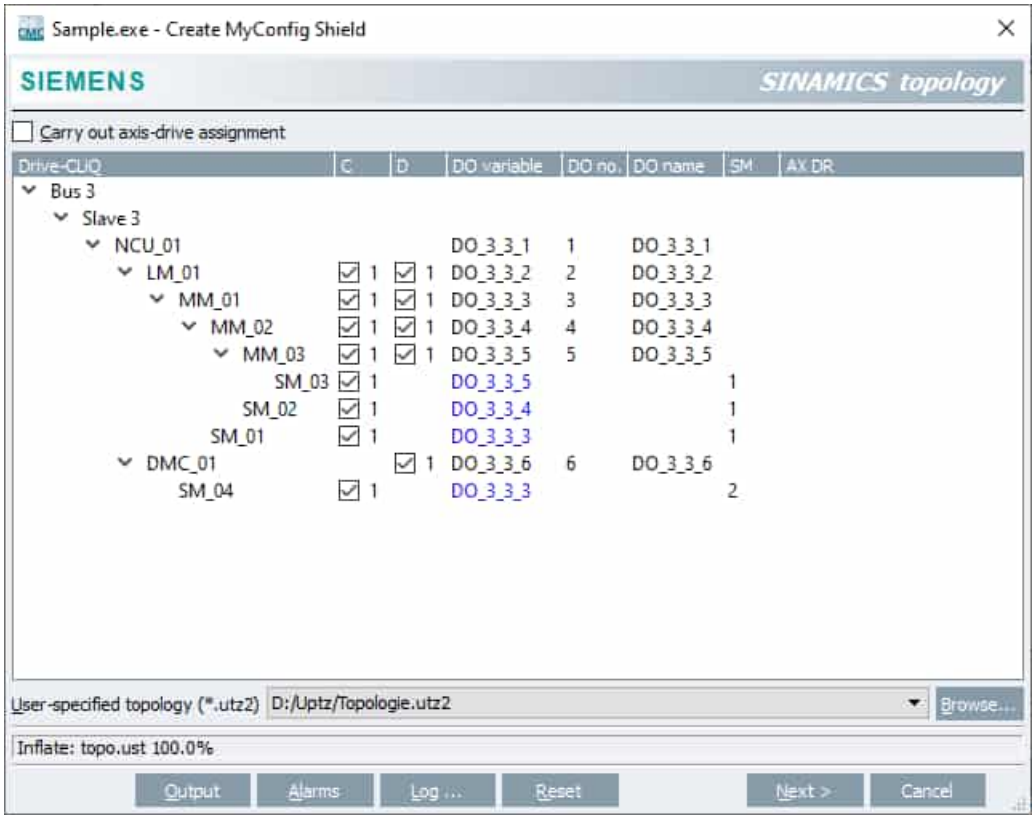


Figure 8-41 SINAMICS topology - commissioning engineer / operator view - example of user-specified topology

Carrying out axis-drive assignment

Setting the check mark activates automatic axis-drive assignment.

As a prerequisite, this Axis-drive assignment must have been configured and activated in Topo.

DRIVE-CLiQ wiring

The DRIVE-CLiQ wiring of the components, the activation state, the assignments of the DO variables and additional properties are displayed in the center of the dialog page.

The 3 different activation states of the SINAMICS elements are:

- "0" for deactivated
- "1" for activated
- "2" for not available

The activation states in columns "C" (Component) and "DO" (Drive Object) can be interactively adapted. Clicking the relevant cell switches through the 3 states. The keyboard can be operated with up/down and letters "K" and "D".

The "SM" column indicates whether a Sensor Module involves a first or second module. Sensor Modules are shown in blue.

Column "AX DR" includes information on the axis-drive assignment.

User-specified topology (*.utz2)

Via the "User-specified topology" drop-down list, a file that was assigned to the package or a file located next to the package (prefix: "./") can be used.

A file from a different location can be selected with the "Browse..." button. You cannot move forward in the dialog without selecting a valid "*.utz2" file.

Manual DO variable assignment

This SINAMICS topology dialog version is used if "Initial state", "Offline archive" **or** "Automatic configuration" was selected **and** no comparison topology was selected for the device configuration of drives.

In this case, automatic axis-drive assignment is not possible using the dialog, as no "Comparison topology (*.ust)" and no "User-specified topology (*.utz2)" are available in which the required axis-drive assignment is possibly configured.

The following diagram is an example of the dialog page when **not** using a comparison topology after automatic device configuration.

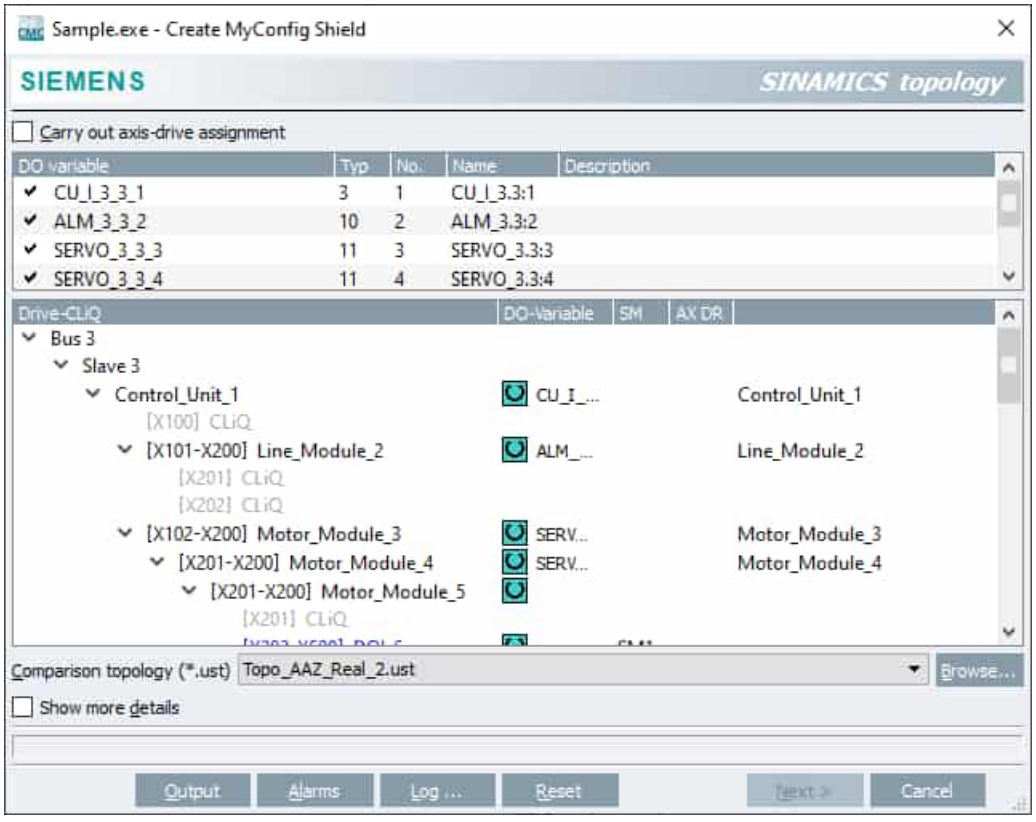


Figure 8-42 SINAMICS topology - commissioning engineer / operator view - example of "manual DO variable assignment"

Precondition for "manual DO variable assignment"

It is only possible to manually assign DO variables if you do not use any "Comparison topology" and any "User-specified topology". You have previously configured a DO variable list in Expert.

The axis-drive assignment must not be activated. This means that the "Carry out axis-drive assignment" check box must not be activated,i.e. no checkmark may be set. The axis-drive assignment is not possible with the "Manual DO variable assignment".

For an interactive manual assignment of the DO variables by toggling to the SINAMICS components, a DO variable must first be assigned to a Motor Module before the DO variable can be assigned to a Sensor Module, so that the sensors can be assigned to axes/drives.

DO variables

A list with all DO variables, which are in the DO variable list, are in the upper part of the dialog page.

The properties of the DO variables are specified in this list, and using a checkmark at the beginning of the line it is indicated whether a DO variable is used, i.e. was assigned to a drive object.

DRIVE-CLiQ wiring

The DRIVE-CLiQ wiring of the components and the assignments of the DO variables to these are displayed in the center of the dialog page.

The "SM" column indicates whether a Sensor Module involves a first or second sensor. Sensor Modules are shown in blue.

Comparison topology (*.ust)

The "Comparison topology" drop-down list remains empty for **manual DO variable assignment**.

Showing more details

More information can be hidden or displayed for diagnostics in the list and tree view using the "Show more details" checkbox.

More information

More information on package execution can be found in the section under "Package execution (Page 401)".

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons".

8.3.22 Version Display (configuration)

This dialog provides an overview of the versions of the control components. The versions can also be fetched in electronic form.

You configure the following property groups for the "Version Display" dialog page:

- Basic dialog settings (Page 220)
- Dialog

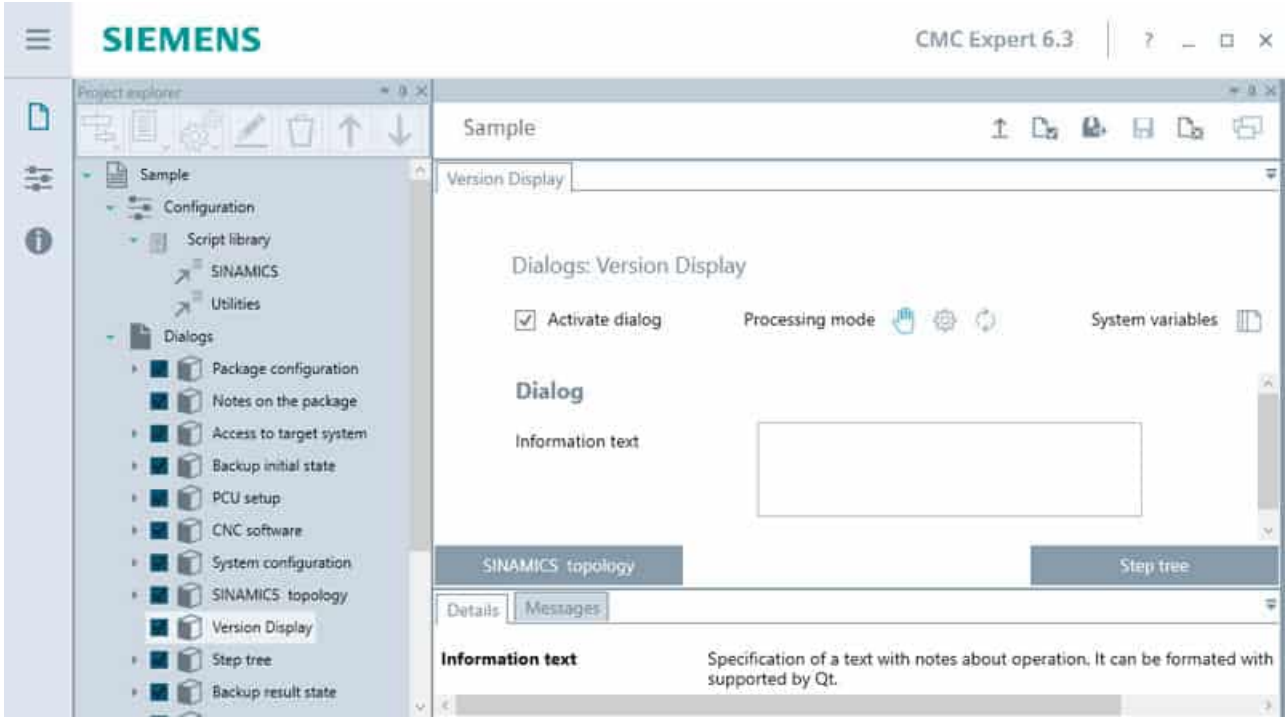


Figure 8-43 Version Display - project engineer view

Note

Specifying data areas

When selecting the "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined.

You can find an overview of this topic in the section "Dependencies of the dialog pages on data areas (Page 240)".

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the Version Display dialog (Page 388)".

Dialog

The following setting is available for "Dialog":

Information text	Insert a text for the operation in the field next to "Information text". Formatting with HTML is possible.

Descriptions of properties / buttons

Properties

The properties for the dialog can be found to the right of the "Project explorer" viewlet in the Home area. If you select a property, the description for this is displayed in the "Details" tab as a separate viewlet below the properties.

Buttons

There are buttons in the Home area below the properties:

- Use the left button to navigate to the previous dialog
- Use the right button to navigate to the following dialog

8.3.23 Version Display (package execution)

The dialog page shows an overview of the versions of the control components. The versions can also be fetched in electronic form.

Dialog view during package execution (for commissioning engineers / operators)

The applications identified during execution and their determined versions are displayed on the "Version Display" dialog page.

The information is obtained via:

- NCU/PPU from the file: "`/siemens/versions.xml`"

More information on package execution can be found in the section under "Package execution (Page 401)".

Note

In the version query (Overview) for the conditional step execution or in scripts, the application names displayed here are used as keywords.

The application names can also be shortened as long as they remain unique.

Information on the step execution can be found in Section "Steps (Page 326)".

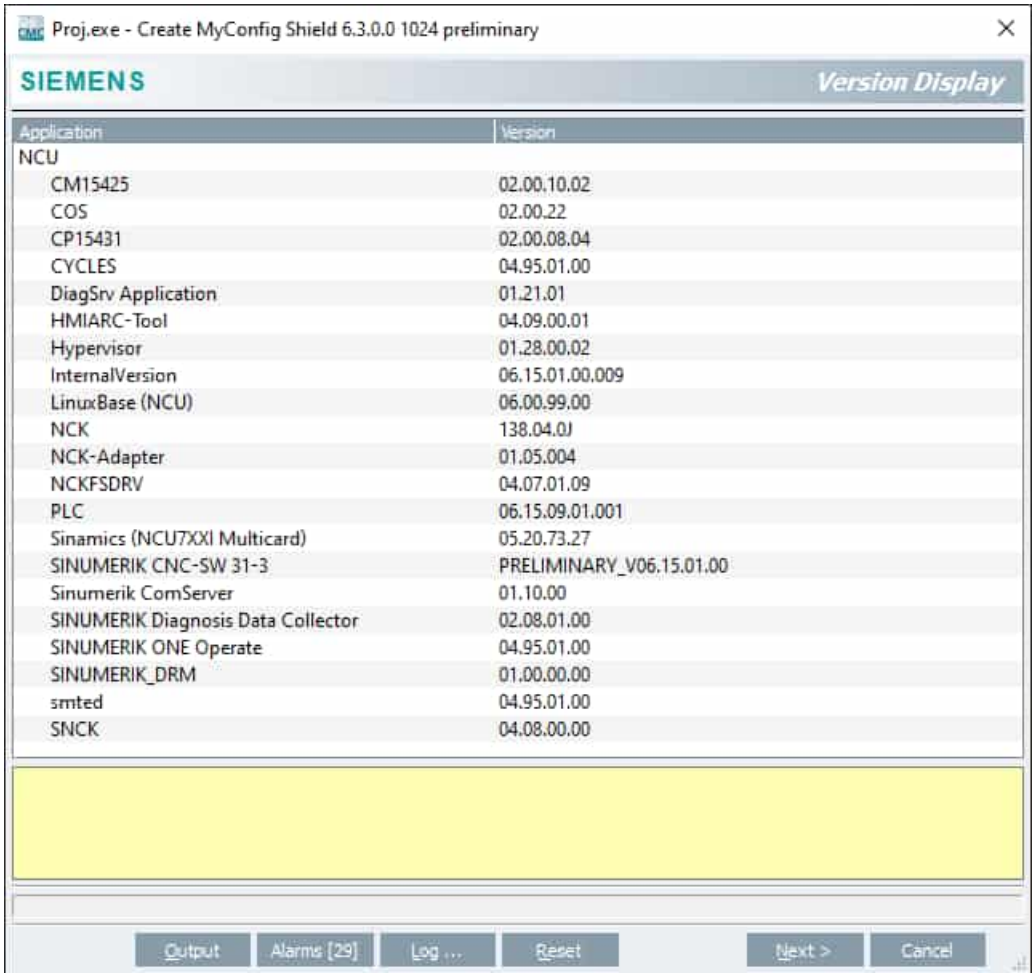


Figure 8-44 Version Display - commissioning engineer / operator view

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons (Page 258)".

8.3.24 Step tree (configuration)

This dialog enables the configuring of the package steps in a hierarchical tree structure.

You can configure the following property groups for the "Step tree" dialog page:

- Basic dialog settings (Page 220)
- Archive

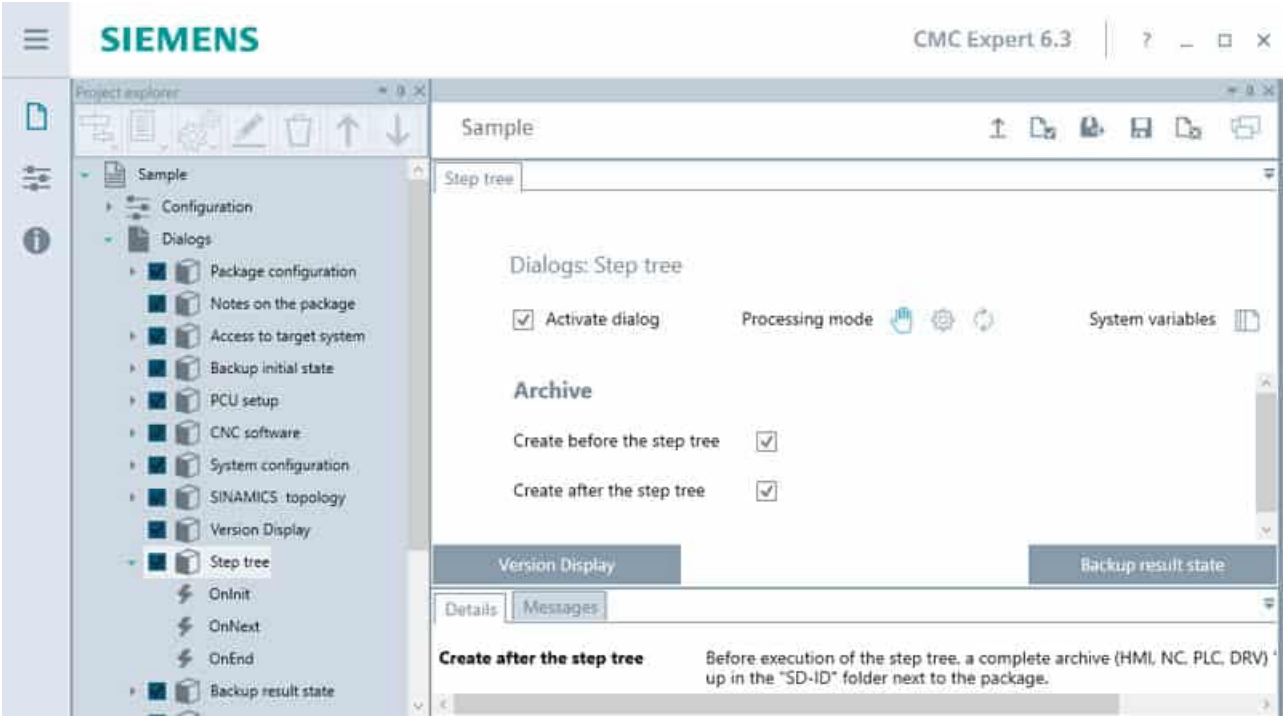


Figure 8-45 Step tree

Note

Specifying data areas

When selecting the "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined.

You can find an overview of this topic in the section "Dependencies of the dialog pages on data areas (Page 240)".

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the Step tree dialog (Page 388)".

Archive

The settings for the creation of archives before and after execution of the step tree are specified by the project engineer. Settings are not displayed during package execution.

You can use the archives for the diagnostics. The archives are not required for the production or the upgrade of a machine.

The following settings are available for "Archive":

Create before the step tree	☐	The check box is deactivated as default setting. If you activate the check box, you define that an archive with the HMI, NC, PLC and DRV areas will be created **before** the execution of the step tree. The archive always includes archive ranges available at the event instant, independent of how the actual use is configured. The backup is performed for the NC area with the setting that all data is to be saved. The archive is backed up with the name `StepBegin_<Date>.dsf`. The archive is stored in the folder named after the "SD card ID" which is located next to the package after the package execution. If no NCU is involved with the package execution, the archive is stored in the "SD-NoneNCU" folder.
Create after the step tree	☐	The check box is deactivated as default setting. If you activate the check box, you define that an archive with the HMI, NC, PLC and DRV areas will be created **after** the execution of the step tree. The archive always includes archive ranges available at the event instant, independent of how the actual use is configured. The backup is performed for the NC area with the setting that all data is to be saved. The archive is backed up with the name "`StepEnd_<Date>.dsf`". The archive is stored in the folder named after the "SD card ID" which is located next to the package after the package execution. If no NCU is involved during package execution, the archive is stored in the "SD-NoneNCU" folder.

8.3.25 Step tree (package execution)

This dialog permits the configuring of the steps in a package in a hierarchical tree structure.

Dialog view during package execution (for commissioning engineers / operators)

On the dialog page, the operator can activate or deactivate steps interactively. The information for the selected step is displayed at the bottom in the yellow field and makes it easier for the operator to understand.

More information on package execution can be found in the section of the same name "Package execution (Page 401)".

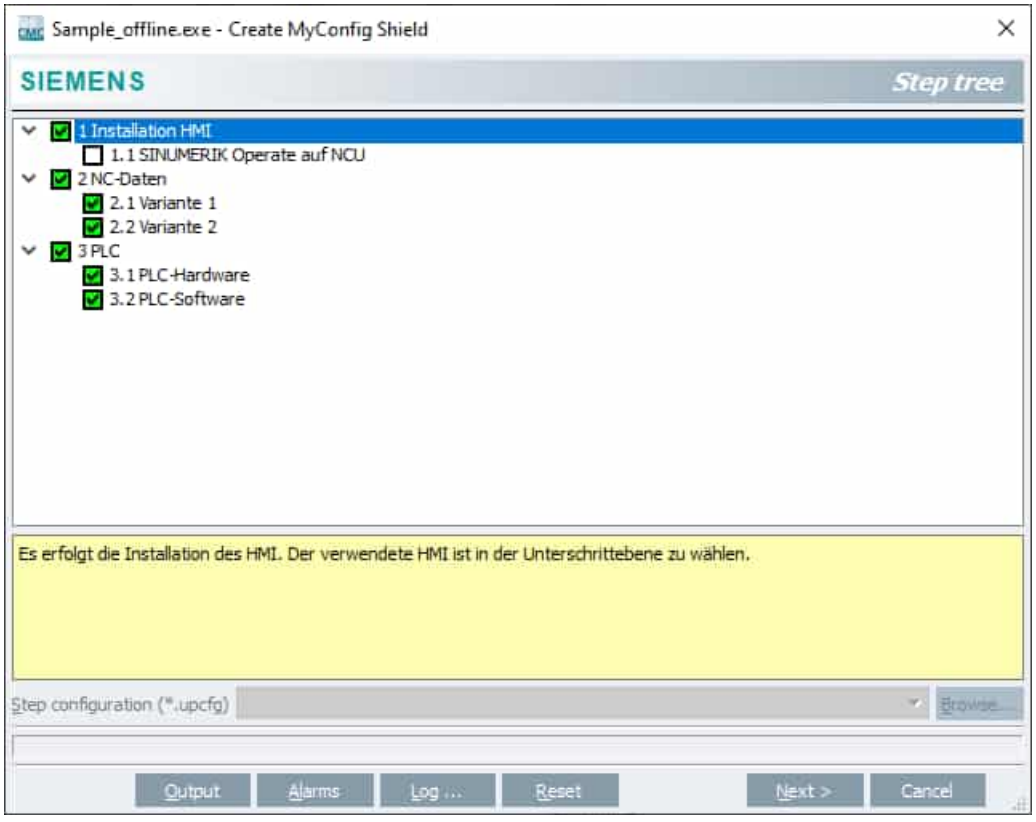


Figure 8-46 1. Step tree dialog - commissioning engineer / operator view

Note

"Next >" button

Clicking the "Next >" button confirms the preparation of the package execution.

"Reset" is then no longer possible.

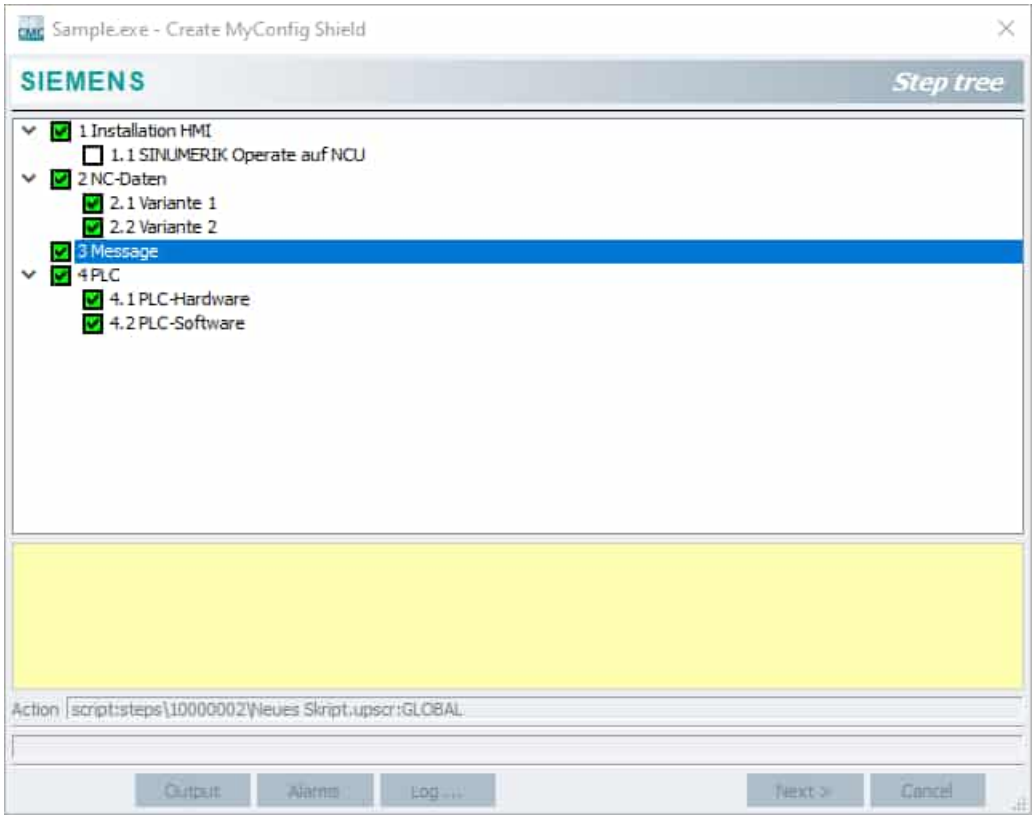


Figure 8-47 2nd dialog "Step tree" - commissioning engineer / operator view

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons (Page 258)".

8.3.26 Archive data transfer (package execution)

While this dialog page is being displayed, the manipulated archive data of NC, PLC and DRV, HMI and SYS is transferred to the control.

The "Archive data transfer" is not listed in Expert because the dialog, without having to be configured, is automatically called from the package if the package data is transferred from at least one of the NC, PLC, DRV, HMI or SYS areas to the control.

The archive data of folder NC, PLC, DRV, HMI and SYS have their source corresponding to what has been configured on dialog page "System configuration" - and are possibly manipulated, removed or added by the package.

All archive data, which is available at the end of dialog page "Step tree" is transferred to the control.

Note

System variable for the "Origin of data"

It is **not** permissible that the system variable for "Origin of data" is at "Do not use", as otherwise archive data is not transferred to the control.

Archive data is also not transferred to the control if the corresponding system variable was set to `UNUSED` when configuring or during package execution.

Dialog view during package execution (for commissioning engineers / operators)

More information on package execution can be found in the section of the same name "Package execution (Page 401)".

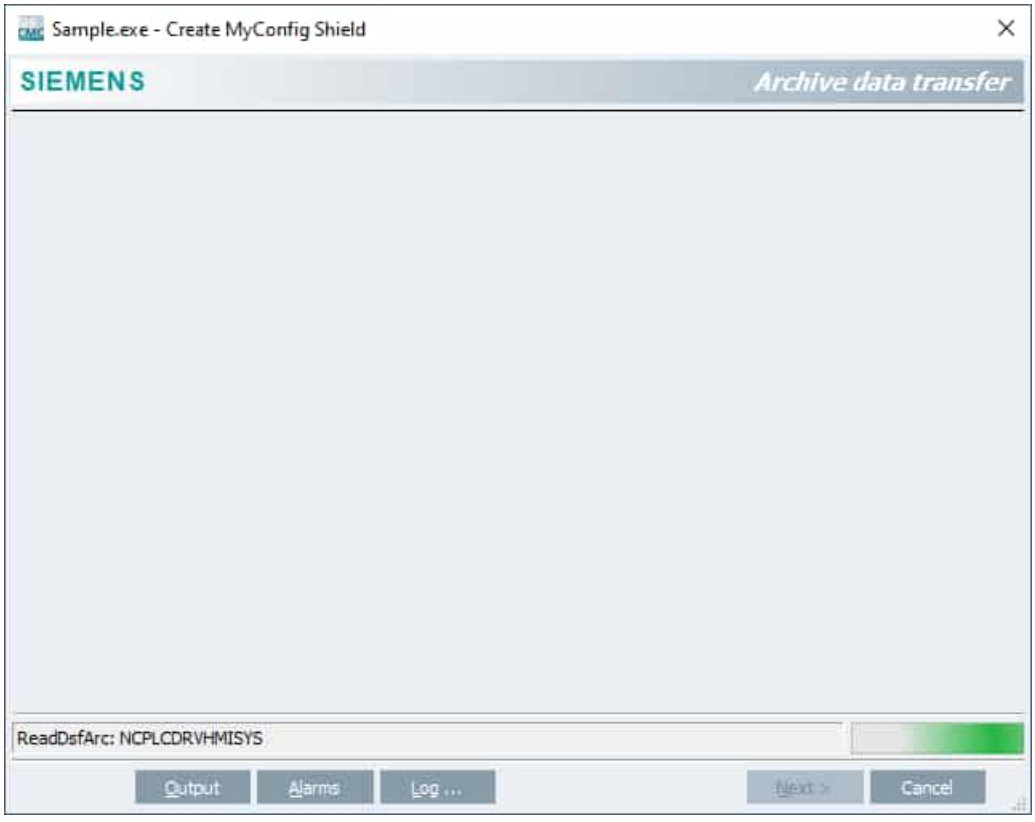


Figure 8-48 Archive data transfer (package execution)

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons (Page 258)".

8.3.27 Backup result state (configuration)

This dialog allows a data backup of the result state of the NCU/PPU after the package execution.

You can save all current control data, including the SD card **or** only the currently active online data of the NC, PLC and DRV subsystems as SINUMERIK archive.

You configure the following property groups for the dialog page "Backup result state":

- Basic dialog settings (Page 220)
- Dialog

- Backup SD card
- Archive

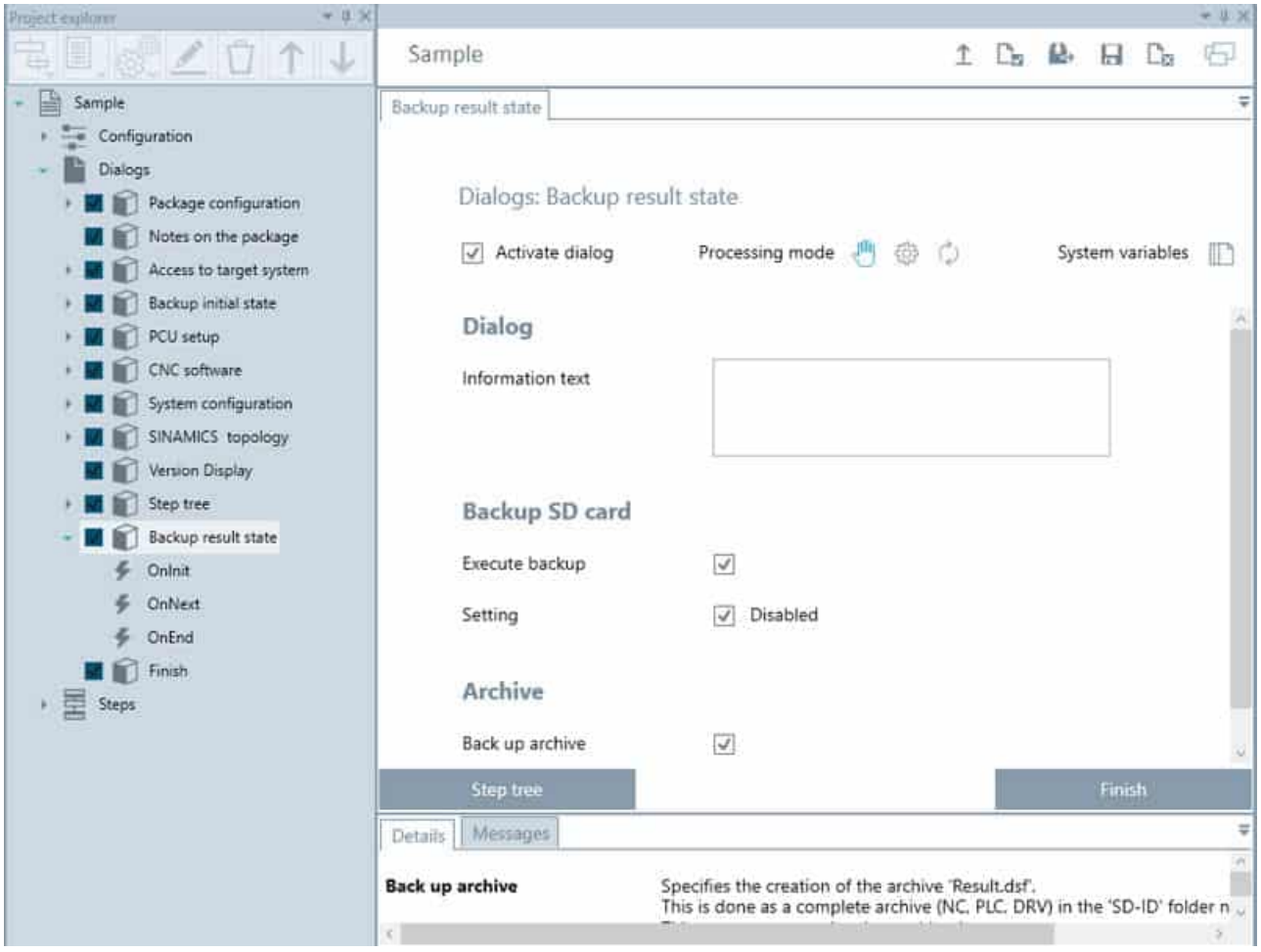


Figure 8-49 Backup result state - project engineer view

Note

Specifying data areas

When selecting the "Data areas" in the "Configuration (Page 236)", the dialog pages that can be used are defined under tab "Dialog".

You can find an overview of this topic in the section "Dependencies of the dialog pages on data areas (Page 240)".

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the Backup result state dialog (Page 389)".

Dialog

The following setting is available for "Dialog":

Information text	Insert a text for the operation in the field next to "Information text". Formatting with HTML is possible.

Backup SD card

Dialog page "Backup result state" can be used to back up all data of the NCU/PPU that runs under the Linux operating system.

The following settings are available for the "Backup SD card" and are activated when the package is executed:

Execute backup	☐	The check box is deactivated as default setting. If you activate the check box, you define that a data backup of the SD card including the archive data for the NC, PLC and DRV areas is created after package execution. The data backup is saved under the name "_result.tgz". The file is stored in the folder named according to the "SD card ID". The folder is located next to the packet after package execution.
Setting	☐ Locked	The check box is deactivated as default setting. If you activate the check box, you specify that the settings made during configuration are blocked. The operator can only view the settings but not change them any more.

Archive

The settings for the "archive" are defined by the project engineer and are not displayed during package execution.

You can use the "Archive" for the diagnostics. An archive is not required for the production or the upgrade of a machine.

The following settings are available for the "Archive":

Back up archive	☐	The check box is deactivated as default setting. If you activate the check box, you define that an archive with the NC, PLC and DRV areas is created **after** the package has been executed. The backup is performed for the NC area with the setting that all data is to be saved. The archive is backed up with the name "Result_<Datum>.dsf". The archive is stored in the folder named according to the "SD card ID". The folder is located next to the packet after package execution.

Descriptions of properties / buttons

Properties

The properties for the dialog can be found to the right of the "Project explorer" viewlet in the Home area viewlet. If you select a property, the description for this is displayed in the "Details" tab as a separate viewlet below the properties.

Buttons

There are buttons in the Home area below the properties:

- Use the left button to navigate to the previous dialog
- Use the right button to navigate to the following dialog

8.3.28 Backup result state (package execution)

This dialog allows a data backup of the result state of the NCU/PPU after the package execution.

All current control data, including the SD card, or only the currently active online data of the NC, PLC and DRV subsystems can be saved as SINUMERIK archive.

Dialog view during package execution (for commissioning engineers / operators)

On this dialog page you select a data backup of the SD card, including all current control data NC, PLC and DRV.

After package execution, the data backup is stored with the name `<SD-card-ID>_result.tgz` in the folder with the SD card ID name that is located next to the package.

More information on package execution can be found in the section of the same name "Package execution (Page 401)".

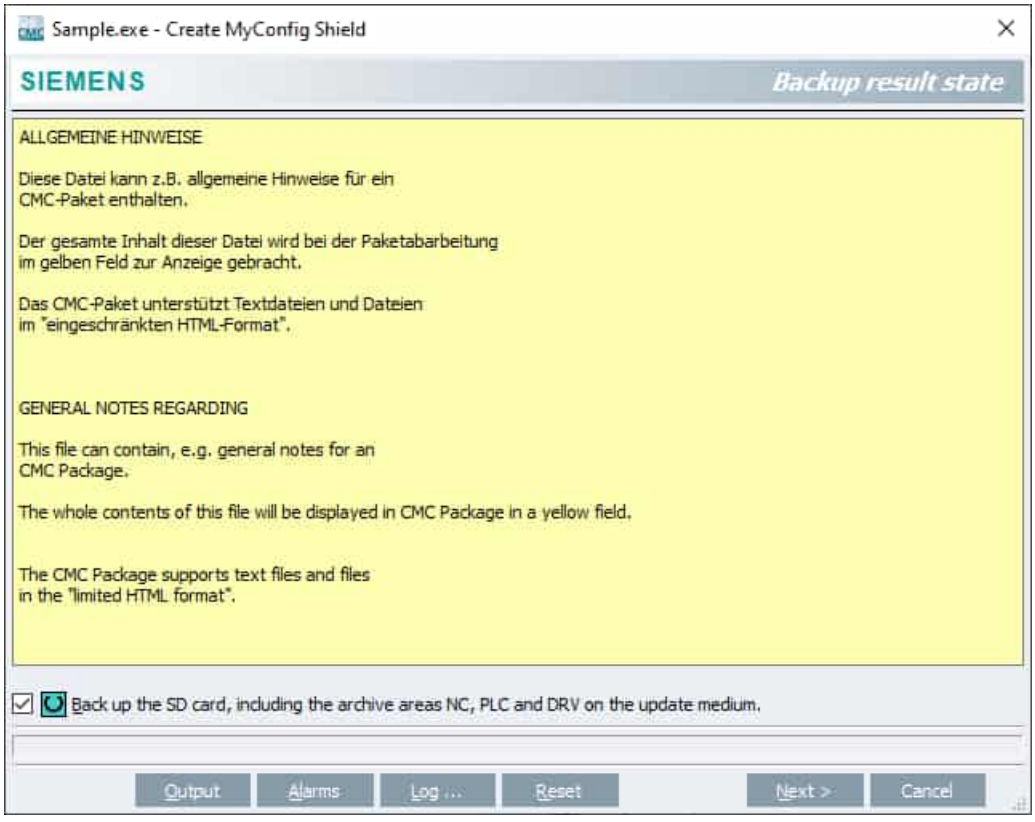


Figure 8-50 Backup result state - commissioning engineer / operator view

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons (Page 258)".

8.3.29 Finish (configuration)

This dialog page completes the package execution and saves the logbook.

You can configure the following property groups for dialog page "Finish":

- Basic dialog settings (Page 220)
- Dialog
- Logbook

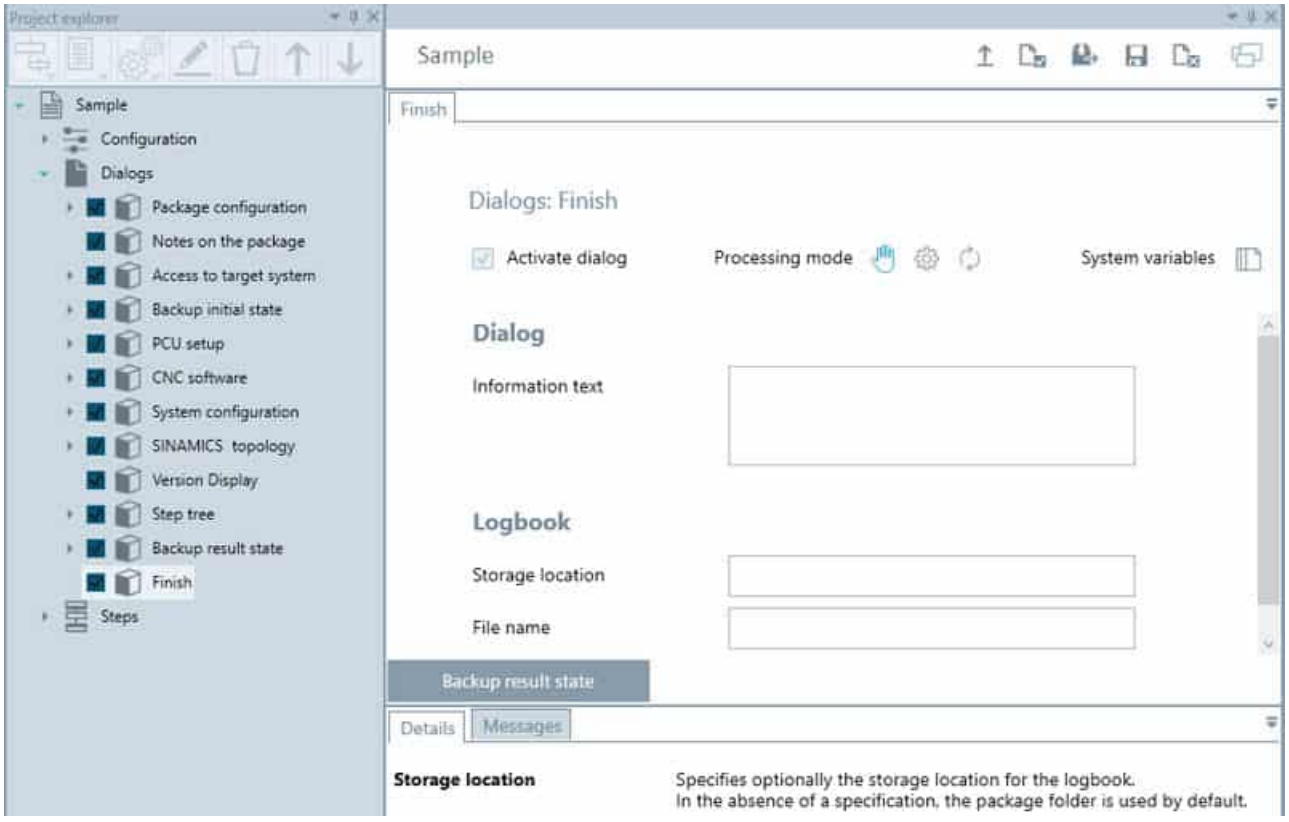


Figure 8-51 Finish dialog

Note

Specifying data areas

When selecting "Data areas" in the configuration page, the dialog pages that can be used are defined in folder "Dialog".

You can find an overview of this topic in Section "Dependencies of the dialog pages on data areas (Page 240)".

Basic dialog settings

Information can be found in the section "Basic dialog settings (Page 220)".

Information on the system variables for this dialog can be found in the section "System variables for the Finish dialog (Page 389)".

Dialog

The following setting is available for "Dialog":

Information text	Insert a text for the operation in the field next to "Information text". Formatting with HTML is possible.

Logbook

The following settings are available for the "Logbook". The settings become effective when the package is being executed.

Storage location	You specify the storage location of the logbook here.
File name	Specify the name of the logbook file without the "*.htm" file extension.

Note

Logbook storage

If a storage location has not been configured, the logbook is stored next to the package.

If a file name has not been configured, the logbook is stored next to the package with the file name "<PackageName>_<Date>.<ext>" in the "<PackageName>_<YYYMMDDhhmm>.<ext>" date format.

The Logbook is saved and archived as "HTM file".

Descriptions of properties / buttons

Properties

The properties for the dialog can be found to the right of the "Project explorer" viewlet in the Home area viewlet. If you select a property, the description for this is displayed in the "Details" tab as a separate viewlet below the properties.

Buttons

The button is available in the Home area below the properties:

- Use the button (on the left) to navigate to the previous dialog

8.3.30 Finish (package execution)

This dialog page completes the package execution and saves the logbook.

Dialog view during package execution (for commissioning engineers / operators)

You cannot deactivate the "Finish" dialog page. A file in text format or restricted HTML format is displayed on this page for postprocessing.

When closing the dialog, the operator can be prompted to save the logbook.

More information on package execution can be found in the section under "Package execution (Page 401)".

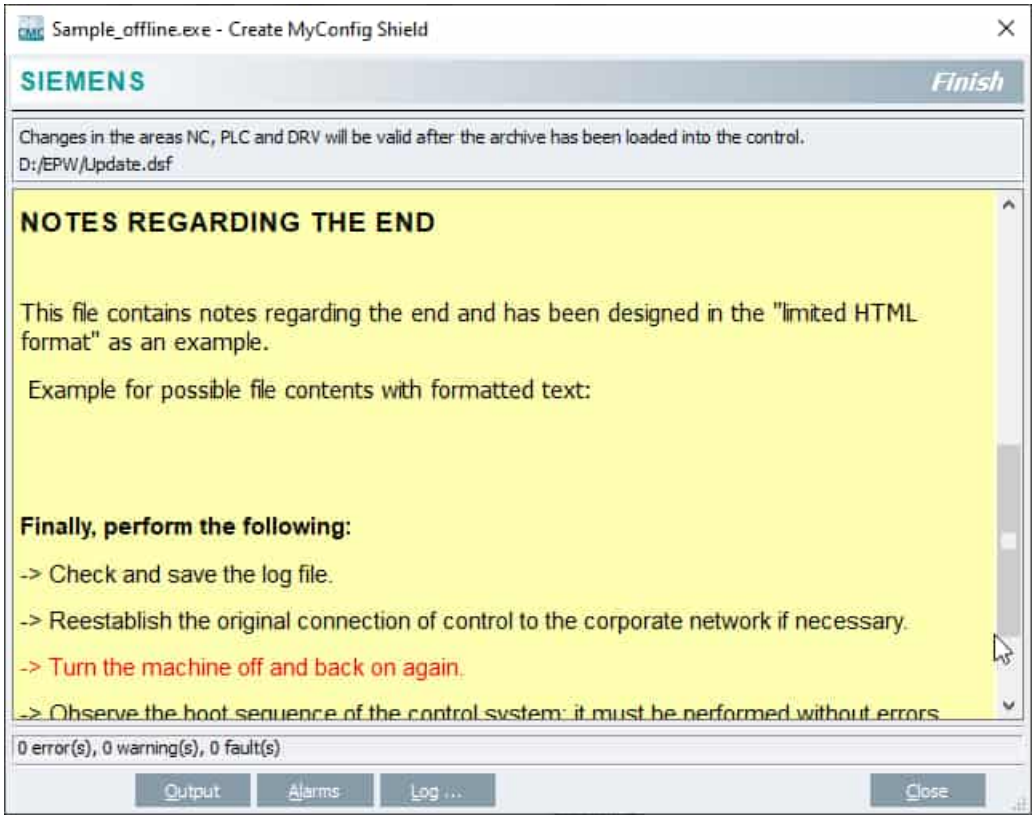


Figure 8-52 Dialog "Finish" - commissioning engineer / operator view

Buttons for the package execution dialogs

Note

The buttons are identical for all dialog pages of package execution and thus have the same meaning.

You can find information on the meanings of the buttons in Shield in the table "Package execution - meaning of the buttons (Page 258)".

8.4 Steps

8.4.1 Overview

You create a hierarchically structured step tree under "Steps".

The step tree is a subtree of the project tree in the "Project explorer".

Structuring of the steps

In the step tree, you can model the different machine variants that belong to a series.

You can create the various machine variants by structuring the steps in several levels - in main steps and substeps. You can activate or deactivate individual steps in the step tree.

Properties are defined for each step during configuration. You can configure the control elements to activate the steps for the "m" out of "n" selection. The step activations that you have specified can also be changed by the operator at the machine while the package is being executed if the step is not configured as "Locked".

Package execution

For package execution, the step tree is executed from top to bottom along the activated check box.

A step is executed when the check box of a step is activated using a checkmark and all higher-level steps which lead to the substep are also activated.

You can see whether a step is actually being executed in the step Properties under the "Step configuration", Property "Execute".

When creating the step tree, arrange the steps in the sequence they should be in when executing the package.

8.4.2 Step tree structure

Step tree structure

The step tree has a hierarchical structure encompassing:

- Steps
- Substeps

- Scripts (Page 336)
- Actions of type
  - "Copy (Page 361)"
  - "Change INI (Page 364)"
  - "Create with preprocessor (Page 368)"
  - "Delete (Page 373)"
  - "Execute (Page 375)"

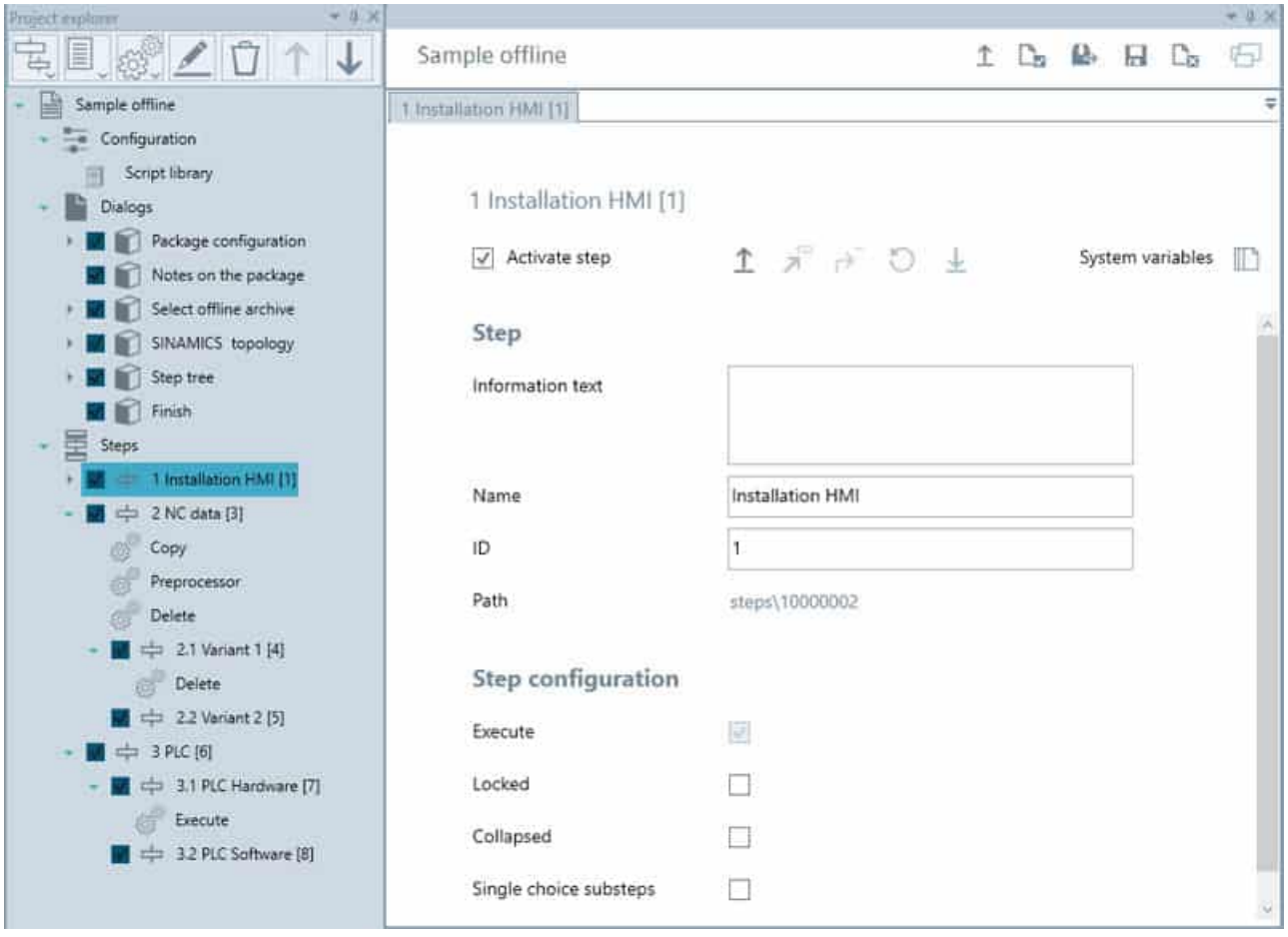


Figure 8-53 Example structure of a step tree

The step name can be freely selected. The step numbering has no functional significance, and is used to improve the overall orientation.

You can paste steps or remove the link using the toolbar. To do this, select the appropriate step. This covers all the properties of a step and its subordinate steps (conditions, component links, notes, etc.).

8.4.3 Edit step tree

Toolbar steps

The functions of toolbar "Steps" is subsequently described.



If you are at the required location in the step tree, you can call additional functions by left-clicking one of the buttons in the toolbar.

Example: Opening functions for step actions



Display tooltip	Move the cursor over the button to display the tooltip.
Open more buttons	You can open additional subfunctions by left-clicking a button. The arrow at the bottom right of the button indicates whether additional subfunctions are available.

Functions in the Toolbar steps

Table 8-18 Toolbar steps

	Step actions		Use this button to paste a new step or substep.
		Paste new step	Pastes a new step after the currently selected step on the same level.
		Paste new substep	Pastes a new substep in the selected step.

	Add script		Buttons for script editing. New scripts are created, already existing scripts are linked. You also create a script to manipulate data via the script.
		Create a new script	
		Link an existing script	
		Link an existing script library	The button has no relevance in the step tree. More information on the script library can be found in Section "Configuration (Page 236)".
	Add action		Create and link an action.
		Creates and links a new action of the type "Copy"	
		Creates and links a new action of the type "Change INI"	
		Creates and links a new action of the type "Create with preprocessor"	
		Creates and links a new action of the type "Delete"	
		Creates and links a new action of the type "Execute"	
		Link an existing action	
	Rename		To rename an object, first select step, script or action and then click on the button.

	Remove from project tree	Select the element that you wish to remove from the step tree. The element can be an action, a script or a step. The element is only removed from the project tree and is still available in the folder of the corresponding step. **Note** When deleting instances, i.e. locally saved elements belonging to the step: This instance is deleted from the project. When deleting links, only the link from the project tree is removed; however, the element remains.
	Move up	Using the arrow keys, you can shift the sequence of selected steps, substeps, scripts or actions that are located on the same level.
	Move down	

Copy and paste step

Keyboard shortcuts allow you to copy or cut a step and paste it elsewhere.

Action	Keyboard shortcut	Description
Copy step	<Ctrl + C>	If the step contains lower-level elements, these elements are also copied.
Cut step	<Ctrl + X>	If the step contains lower-level elements, these elements are also cut.
Paste copied or cut step on the same level	<Ctrl + V>	The step is pasted on the same level after the selected step.
Paste copied or cut step on the same as an substep	<Ctrl + Shift + V>	The step is pasted as a substep of the selected step.

8.4.4 Step properties

You can configure the following property groups for each step or substep:

- Basic step settings / Step project
- Step
- Step configuration

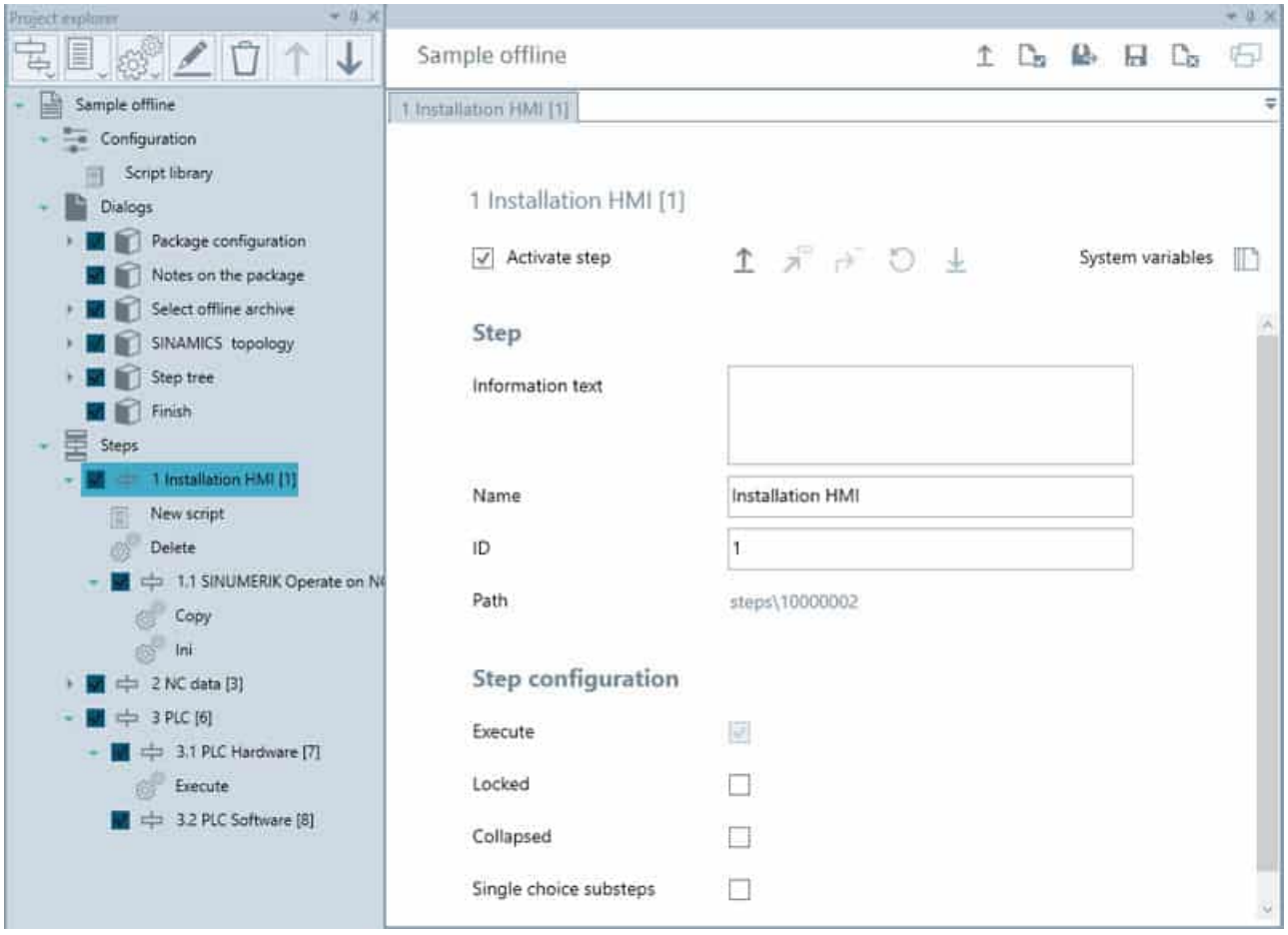


Figure 8-54 Step properties

Basic step settings / Step project

Table 8-19 Basic step settings / Step project toolbar

☑	Activate step	With this check box, you specify whether or not a step is used for package execution. This step will only be executed if you have activated the check box and all the higher-level steps associated with it are also activated. Activate a step using a checkmark in the check box. If a step is not activated, this step - and if they exist - all lower-level steps associated with it are not executed.
	Step project toolbar	The Step project toolbar is used to export step projects from the step tree or link them into the step tree of an Expert project. More information is provided in Section "Step project (Page 338)".

		Exporting a step project from a selected step	In the step tree, select a step from which you want to export a step project. All substeps, scripts and actions are also exported. Links are exported as links. If you have configured links in the exported subtree, then you must ensure that these links are also valid links in the target project (location where they are used in the step project).
		Linking a step project	In the step tree, select a step as the connection point from where you wish to link a step project. After linking the step project, the connection point is displayed in the step tree with blue text.
		Removing a link to the step project	In the step tree, click the step highlighted in blue to remove the linked step project from the step tree.
		Refresh	Click "Refresh" to receive changes from the external step project in the linked step project.
		Importing step attributes from the step project	Attributes that can be edited from the external step project are accepted in the linked step project of the Expert project.
		System variables	Left-click on the icon for system variables to open the window in which the system variables for the steps are listed. For more information on the system variables, see "Reference list system variables", section "System variables for the steps (Page 393)".

Step

The following settings are available for the property group "Step":

Information text	You can enter information text for each step; this text is then displayed on the "Step tree" dialog if the particular step is selected or being processed.
Name	You can freely name each step. The name has no functional meaning. However, the "step name" can help the experts working in Expert and the operator at the machine with regard to content and meaning of the step.

ID	The ID can be subsequently modified. Either numeric values or alphanumeric values starting with a letter or '_' are permitted. While executing the step, the activation state of a step in scripts can be queried and/or controlled using its ID. **Note** **Step project** The IDs in a linked step project cannot be edited; however, they can be referenced. Referencing a step in a linked step project is possible using the ID, just the same as for normal steps. The special feature of the ID of the linked step project is that the ID comprises 2 parts. The meaning of the ID: • 1st position is the connection location in the step tree • 2nd position is the ID of the step from the external step project For example: `Up.$Step[2.6].Activated = True`
Path	Specifies the step path. The unique ID used in the path comprises 8 digits and cannot be changed. Marking "upsteps" at the start of the path indicates that the step belongs to a linked step project.

Step configuration

The following settings are available for the property group "Step configuration":

Execute	☑	If a step is activated, the "Execute" property is automatically activated and can no longer be deactivated. The step is executed during package execution. If a step is either not activated or not executed due to a missing activation in a deactivated higher-level step, then "Execute" is automatically displayed as deactivated. This step is not considered during package execution.
Locked	☐	If a step is locked by the configuration engineer, the operator can no longer activate or deactivate the step during package execution. The selected step is locked and the operator cannot see it.

Collapsed	☐	For the time of package execution, you can define for each step whether the step will be collapsed. Substeps are thus hidden and are not visible to the user. You can thus configure the overview for the display of the step tree individually. In the default setting, substeps are displayed while the package is being executed. All substeps are hidden if the check box setting is activated.
Single selection substeps	☐	The steps of a level can either be selected or deselected a multiple number of times or individually. **Multiple**: All steps can be selected/deselected independent of one another. **Individually:** Selecting one step deselects all other steps. By activating the checkbox, the substeps of a selected step become options. You can select only one option or variant.

Note

More information

Information on the system variables, see "Reference list system variables", in Section "System variables for the steps (Page 393)".

8.4.5 Script properties

You can configure the following property groups for every script:

- Script
- Execute

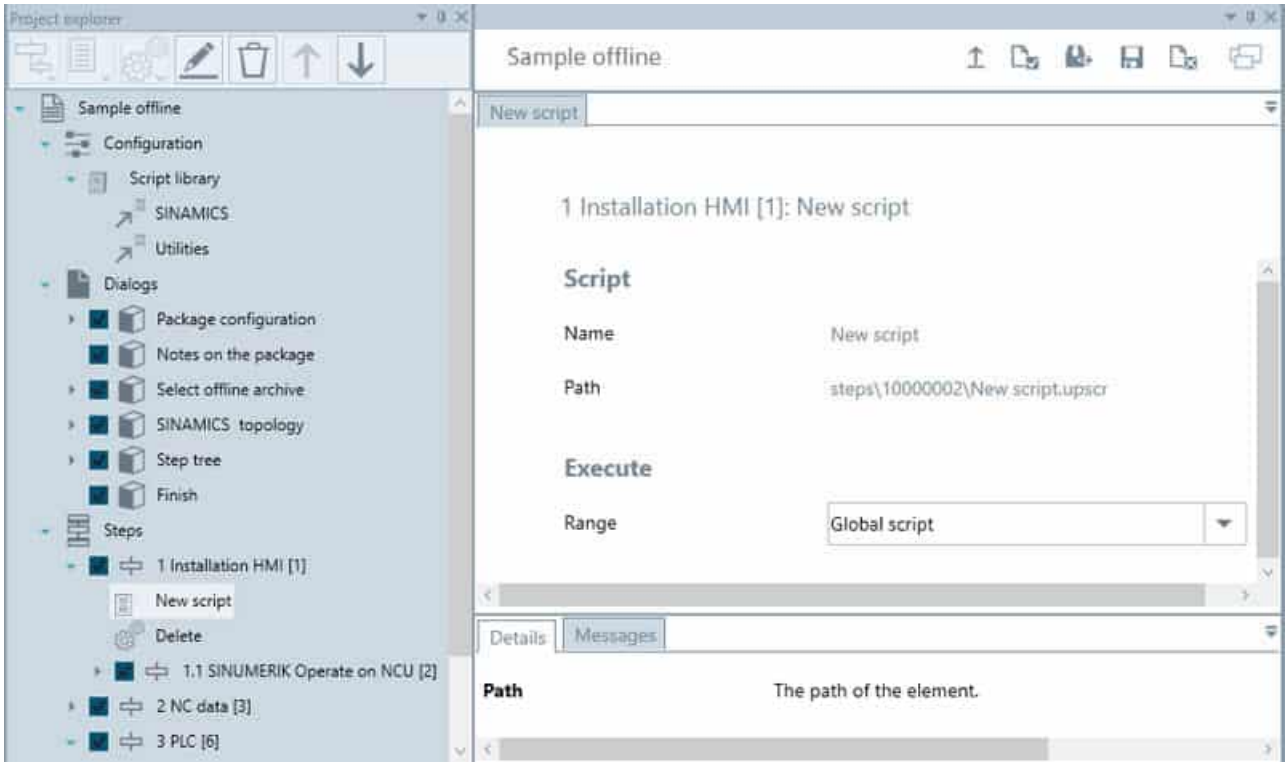


Figure 8-55 Properties "New script"

Script

The following settings are available for the property group "Script":

Name	New script	Display of file name
Path	steps\10000001\New script.upscr	Display of the path of the linked script. Display where the script is saved. The path is specified relative to the project.

Execute

The following settings are available for the property group "Execute":

Area	NC SINAMICS Global script	Select the area in which the script should be executed. "Global script" is the default setting.

8.4.6 Scripts

You can save scripts for each main step or substep in the step tree.

To do this, select the appropriate step and paste a script via the toolbar.

The following options are available:

Table 8-20 Script options

Button		Meaning
		Open script options to add a script
		Create a new script
		Link an existing script

A script is always assigned to a step. A script can include manipulation instructions, for example.

Using the integrated "Script Editor (Page 29)", you can open and edit the newly created or linked scripts.

Executing steps and scripts

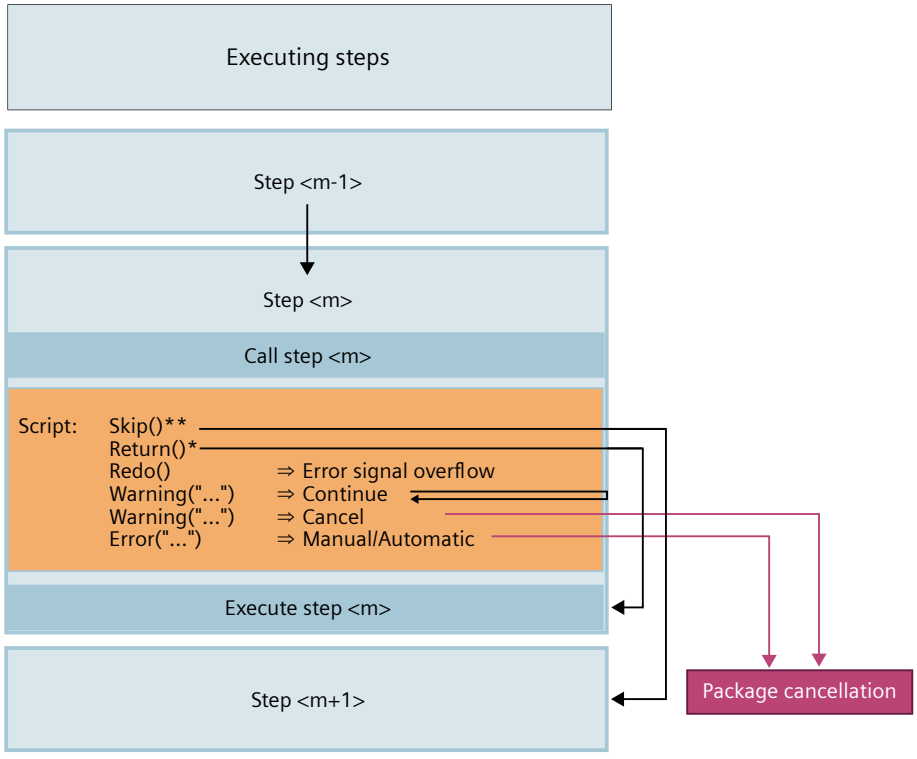
The execution of steps and possibly existing scripts (shown in colored boxes) is schematically shown in the following diagram.

Procedures are listed as example, that can potentially control the execution sequence of the step and/or script.

In principle, you can use all script language elements in the script.

The shortcut menu supports you when generating your script.

More information on the script language is provided in the Section under "Create MyConfig CMC script (Page 29)", taking into consideration of the constraints in Section "Data access (Page 50)".



* Return() jumps to next step. This can either be a main step or a substep.

** Skip() jumps to next step in the same level. From one main step to the next main step, from one substep to the next substep. The current step is no longer executed.

Figure 8-56 Executing the script for steps

8.4.7 Step project

8.4.7.1 Overview

A step project is part of a step tree that is exported externally so that it can be linked to one or several projects. The step project file "*.upstep" can contain any number of substeps, scripts, and actions. The configuration engineer exports the step project and thus provides the user with a subtree.

The step project has the following advantages:

- The configuration engineer creates the subtree once
- Central archiving and management
- Centralized provision, independent of a CMC project
- Multiple use for recurring step sequences, e.g. for:
  - Operator panels
  - Tool magazines
  - PLC components

8.4.7.2 Exporting a step project from a selected step

When commissioning machines, sometimes several projects must be managed.

To support users, a subtree can be exported from a step tree. This subtree, designated as "Step project", is externally archived from an Expert project. Customers create the step project, and this is managed at a central location.

Procedure

In a step tree of an Expert project, you have created a step with substeps, scripts and actions. You require this subtree in this or other projects for step sequences that are being continually used. To simplify the procedure, export this part of the step tree as autonomous "Step project".

1. In the Project explorer, select the step from which you wish to export the step project.
2. Click on "Export step project from selected step".
   The target directory opens.
3. Define the file name (*.upstep), and save the step project in an empty target directory.
   You can already create and name the target directory on any drive beforehand.

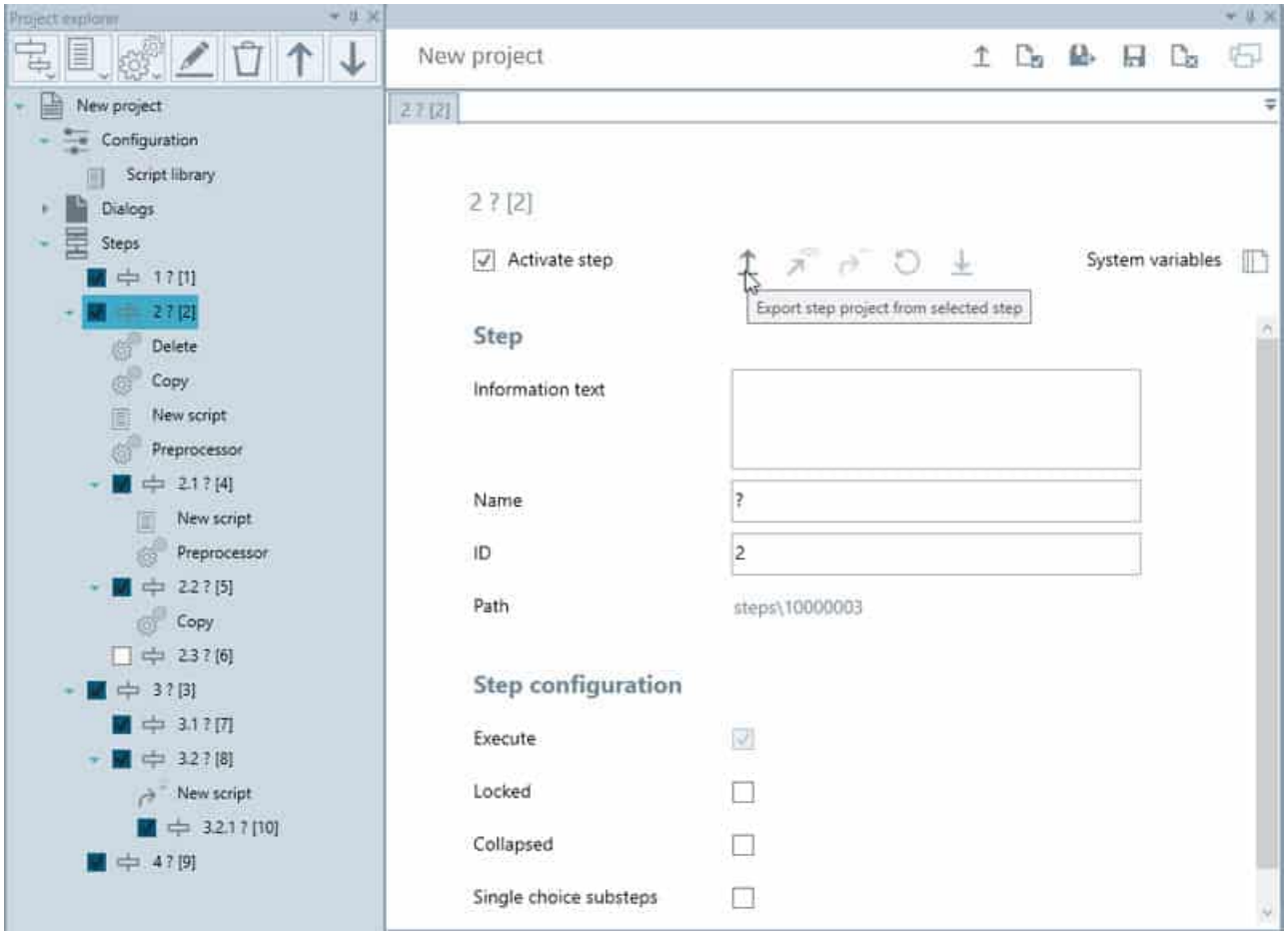


Figure 8-57 Exporting a step as external step project

Note

Target directory for the step project

The target directory for the step project must not contain any other data. Only an empty folder or a folder with an existing step project is allowed. An already-existing step project will be replaced.

Folders containing other files are not emptied.

8.4.7.3 Linking a step project

An "Expert project" centrally archives and manages "Step projects" that have been exported externally. This means that the step project can be used for any project or linked to a project.

As a consequence, a step project is exported from a master project. An external step project is linked to any target project. The external step project linked to the target project is then a part of the target project. The linked step project brings with it attributes from the master project. You can change attributes in the target project. This means that changed attributes become local attributes, which are independent of the step project of the master project.

Procedure

Proceed as follows to link an external step project to an Expert project:

1. Select the connection point in the Project explorer at which you wish to link the step project. The connection point must always be an empty step.
2. Click on "Link step project".
3. In the opened directory, navigate to the step project file (*.upstep).
4. Select the file and confirm with "Open".
   The previously selected connection point is now marked in blue in the Project explorer.

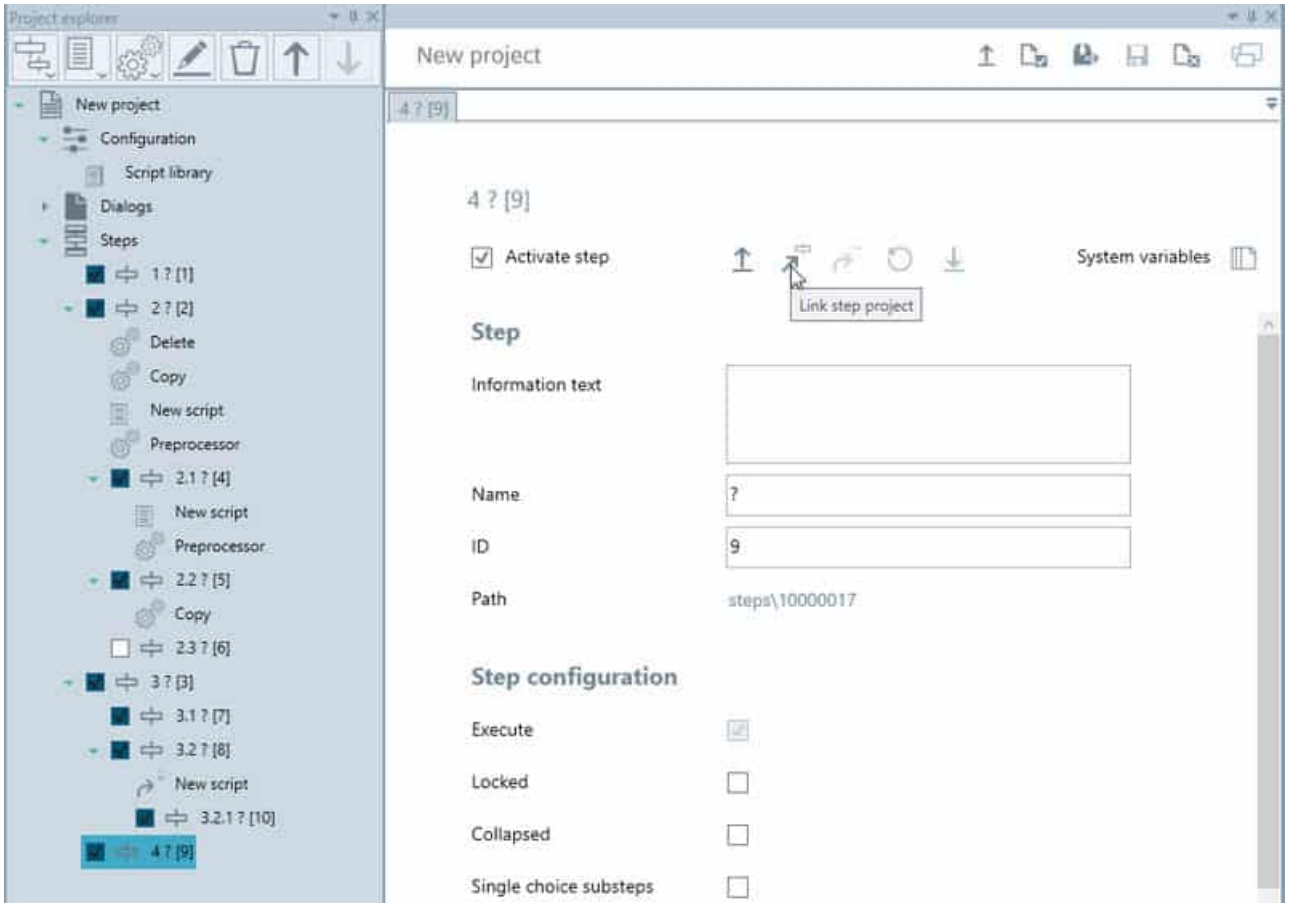


Figure 8-58 Linking a step project

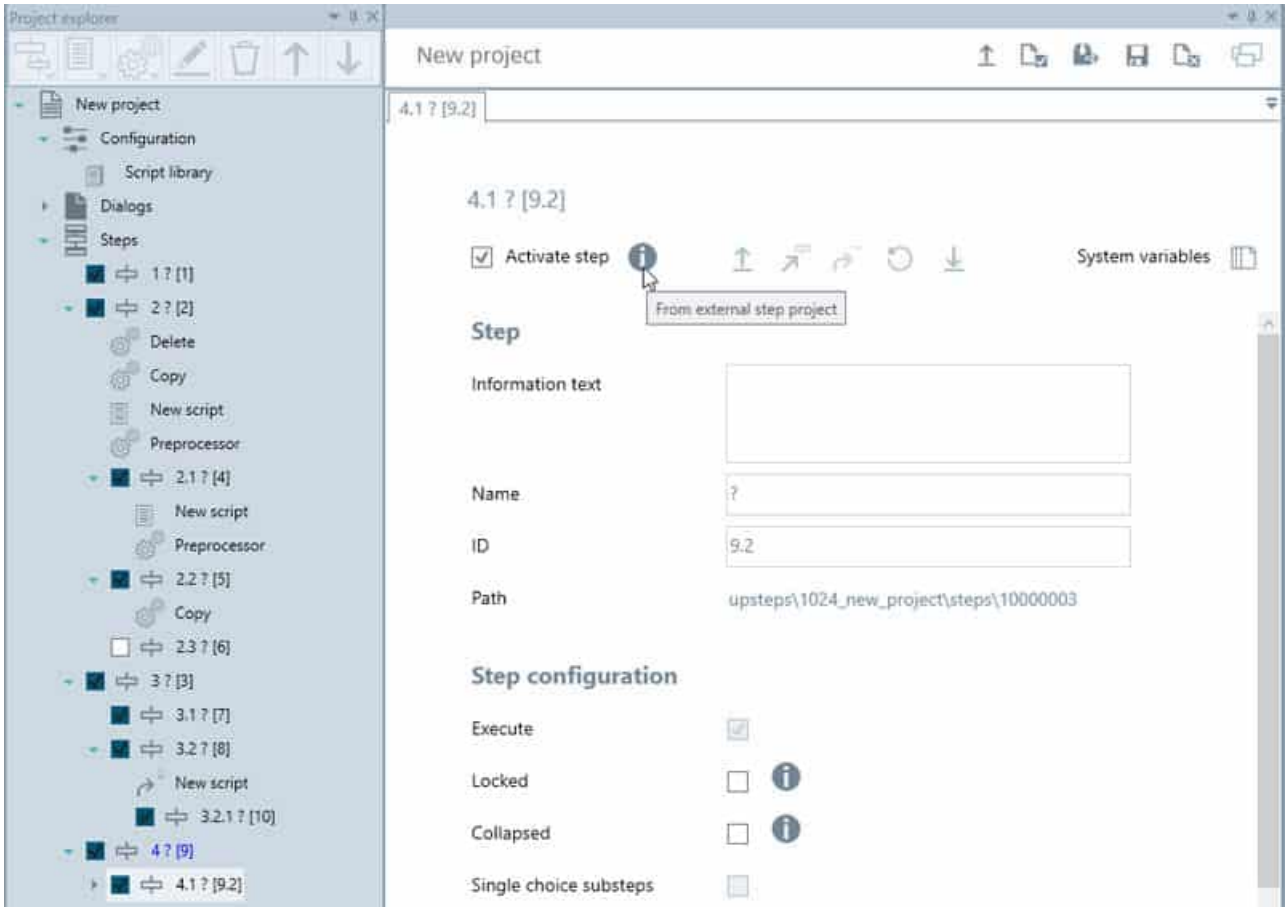


Figure 8-59 Diagram of a linked step project

Diagram showing a linked step project

Features of a step project:

- The connection point is shown in blue in the Project explorer
- The step project is assigned its own ID
  ID [9.2] in the main step of the step project shows:
  - "9" as ID for the linkage position
  - "2" is the original ID of the step in the external step project

  The IDs of the steps in a step project cannot be changed.
- Actions and scripts contained in them are shown as link
- Icon "From external step project" is used to display the attributes that can be edited, which are currently included from the external step project.

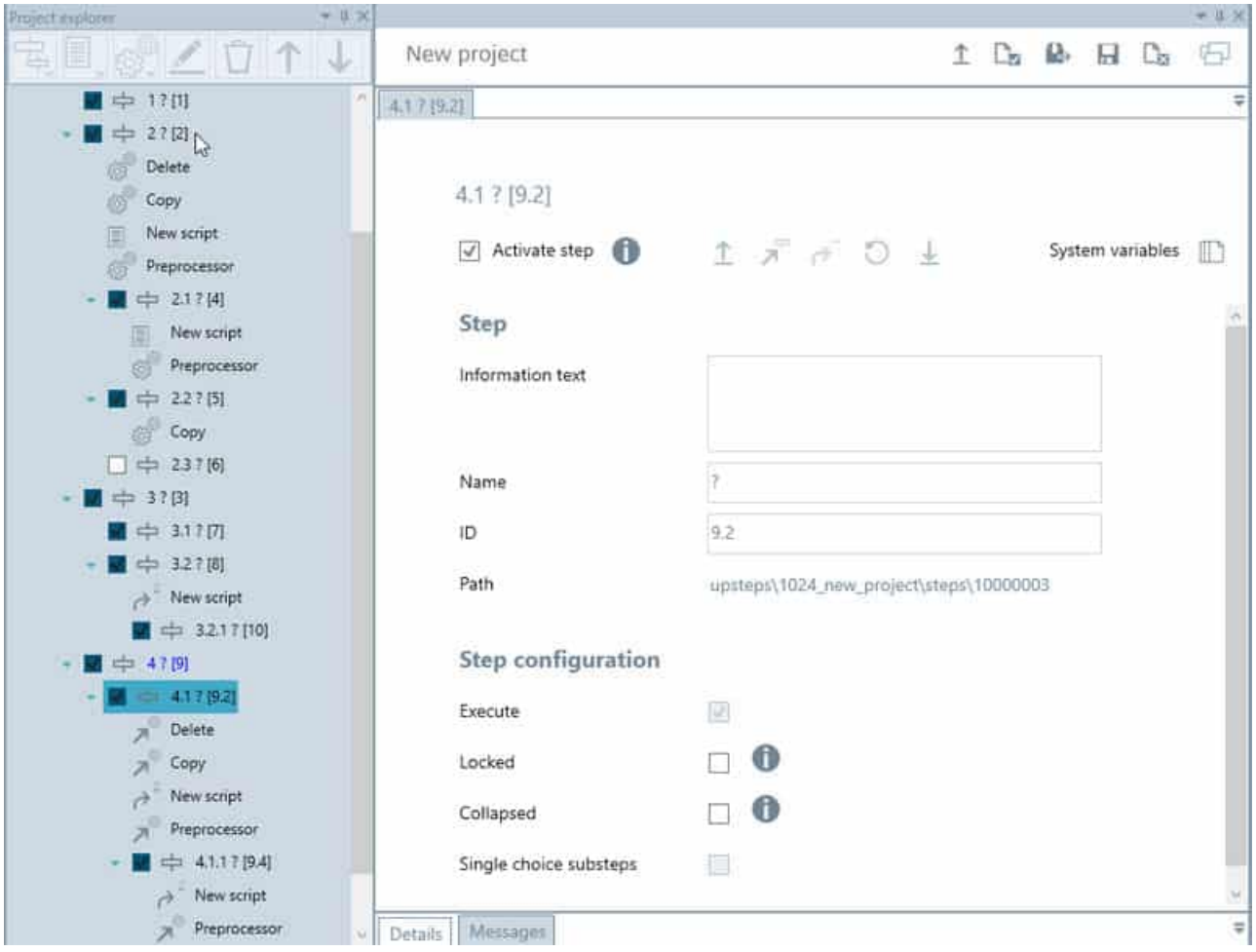


Figure 8-60 Step project - compare exported subtree with linked step project

Properties of the step project

The same step project can be linked at several locations in an Expert project.

After being linked to an expert project, a step project can be adapted to address individual requirements. Only properties marked as attribute "From external step project" can be changed in the linked step project.

If an attribute in the linked step project is changed, then the change is applicable for the modified location. As a consequence, the changed value replaces the value from the external step project at the current location and deletes attribute "From external step project". This means that the changes that come from the external step project are no longer updated (refreshed) for this changed property as a local definition exists for the property.

If changes should be made to the linked step project, e.g. a new action, new scripts or substeps from the external step project, click on icon "Refresh".

8.4.7.4 Removing a link to the step project

You can remove a link to a step project.

The step project can only be deleted from the Expert project from the connection point as complete subtree. The attributes that you locally changed in your target project from the external step project are also lost.

In the case of a new link, you can only link the external step project of a master project. Further, you must subsequently adapt the local attributes that you require.

Procedure

Proceed as follows:

1. Select the step project at the connection point marked in blue.
2. Click on "Remove link to the step project".
   The step project is deleted from the Expert project.
3. Save the project.

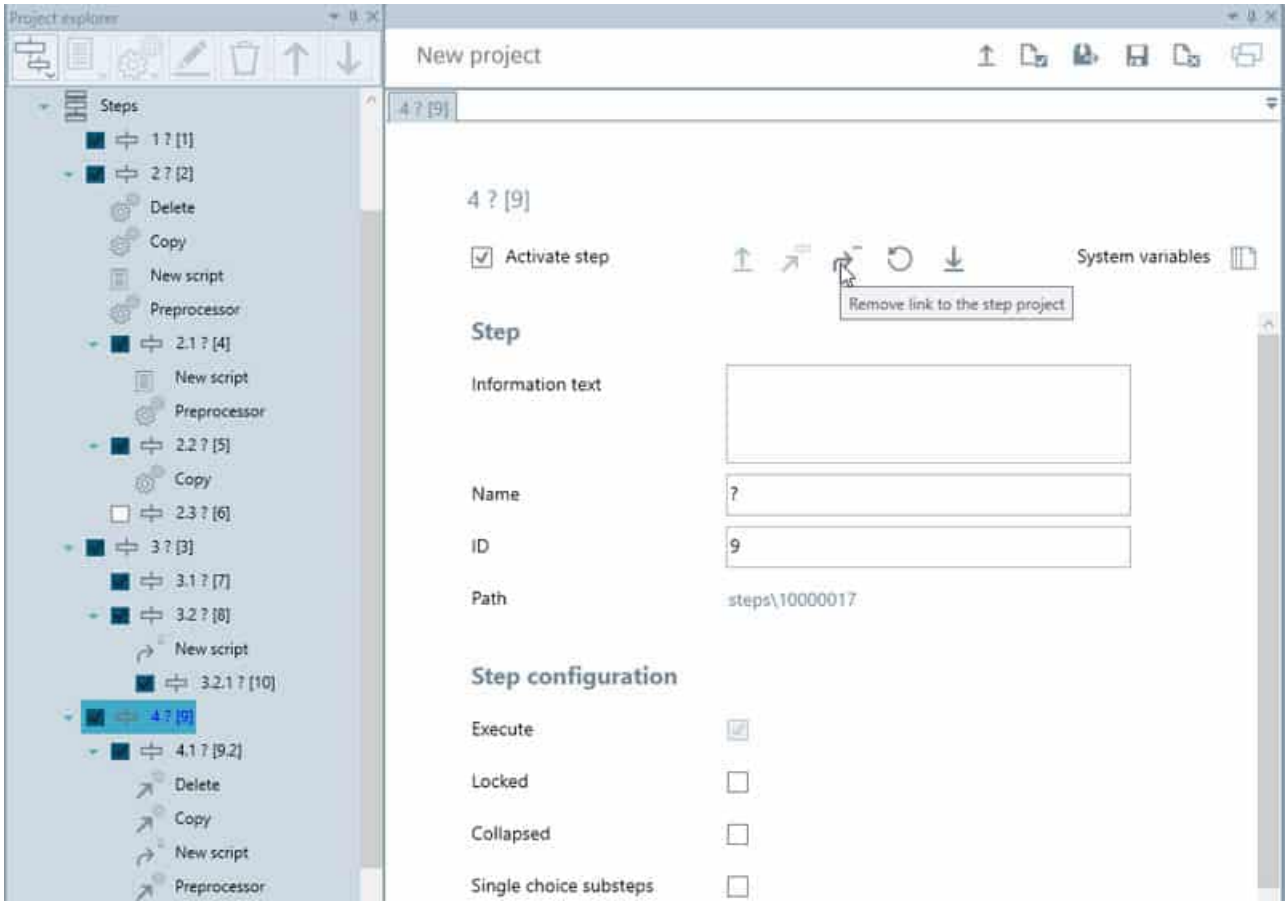


Figure 8-61 Removing a link to the step project

8.4.7.5 Refreshing the step project

A step project exported from a master project was extended by one or more elements, e.g. substeps, scripts and/or actions.

To set the linked step project in your target project to the current status of the external step project, refresh the linked step project using the appropriate button.

Procedure

Proceed as follows:

1. To refresh, select the linked step project at the connection location.
2. Click on button "Refresh".
   The linked step project contains all updates from the external step project.
   Attributes, which were already changed in the linked step project, are local attributes and are therefore no longer overwritten when the button is pressed.

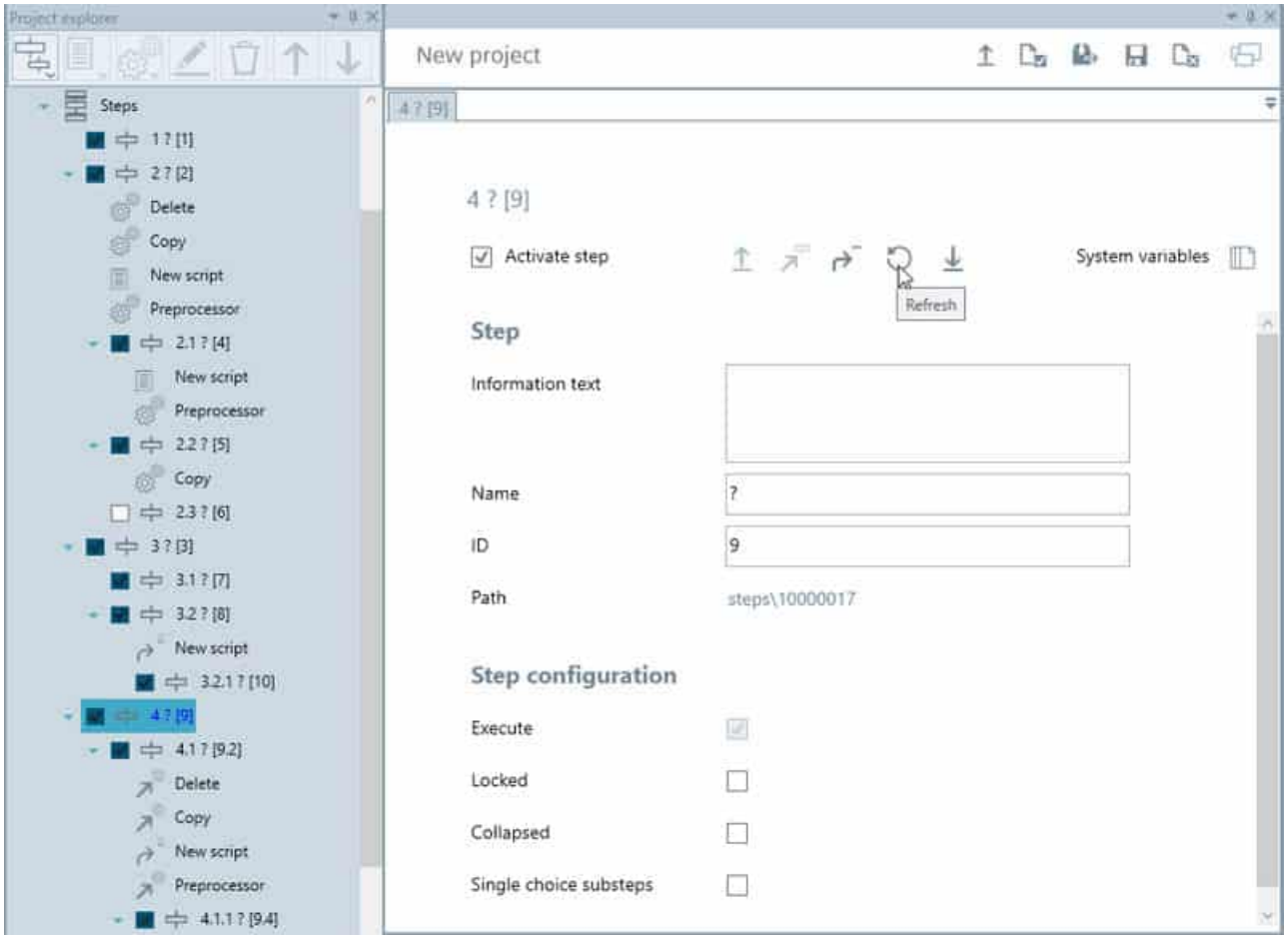


Figure 8-62 Refreshing a linked step project

8.4.7.6 Importing step attributes from the step project

Using this button, attributes that were edited in the external step project in the linked step project are updated in the linked step of the Expert project Attributes already existing in the project are overwritten by the external step project. From this point in time, this data is part of the project and can be freely modified.

Updating the step project no longer changes these attributes. When updating, the attributes of new steps in the step project are shown as linked attributes just the same as for the linking ("From external step project").

To also accept these attributes in the project, initiate a new import of the step attributes, therefore overwriting local deviations. Or you must manually edit the attributes of the new steps, therefore overwriting them.

Procedure

Proceed as follows:

1. To import attributes from the external step project, select the linked step project in your Expert project at the connection location shown in blue.
2. Click on "Import step attributes from the step project".
   The linked step project is assigned the attributes from the external step project.
   All of the changed local attributes are overwritten, and reassigned the attributes from the external step project.

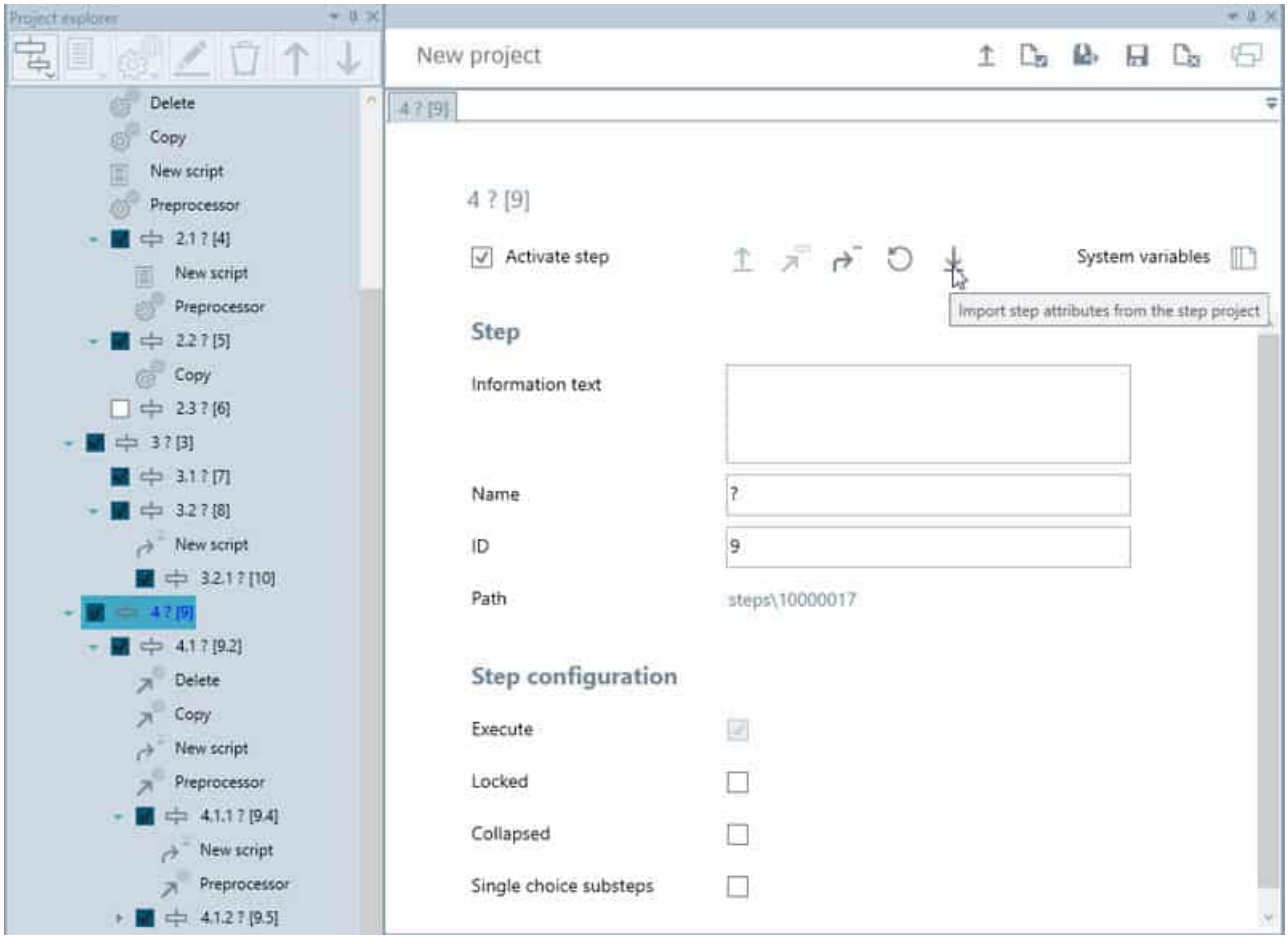


Figure 8-63 Importing step attributes from the step project

8.4.7.7 Command line call

Step project

To export a step from an Expert project in the step tree as an external step project, you can start Create MyConfig Expert in command line mode.

To export in the command line mode, specify command line parameters (switches), which are in some instances optional.

Call

Write the command subsequently shown as well as all associated switches and arguments in one line to a batch file or a command line interface.

Exporting the step project

```
ExpertCmd.exe exportstep -s <StepId> -o <Target path> [-c] [-x <log
file path>] <project file path>
```

The switches in square brackets are optional.

Example 1:

```
ExpertCmd.exe exportstep -s 2 -o D:\td\upstep\part2.upstep -c d:\td
\master3\master3.upproj
```

Example 2:

```
expertcmd exportstep -s 1 -o
```

```
d:\Projects\CMC\10_ExpertProjects\99_Data\08_step_projects
\01_DeviceTypeQuestion\Abfrage_Geraetetyp.upstep -c -x
```

```
d:\Projects\CMC\10_ExpertProjects\01_TestProj_StepProj_Master
\exportlog.xml
```

```
d:\Projects\CMC\10_ExpertProjects\01_TestProj_StepProj_Master
\StepProj_Master.upproj
```

Notes

Note the following information for the command line call:

Note

Specification of the complete path of "ExpertCmd.exe"

Specify the complete path of the "ExpertCmd.exe", or include the path in the environment variable "Path".

Example:

```
"C:\Program Files\Siemens\Automation\SINUMERIK\Create MyConfig
6.3\ExpertCmd.exe"
```

Note

Path data

- Set the specified paths in straight quotation marks if spaces are included.
  Example:
  `"C:\path with spaces\Expert_Logfile.xml"`
- When specifying folders, path data may only end with `'\'`, e.g. `-d "D:\packages\"`

Note

Saving batch files

Always save batch files with ANSI/ASCII coding.

Note

Umlauts and special characters

Avoid the use of umlauts and special characters in path names.

To allow special characters or umlauts to be correctly processed in specified paths, save the batch files with command line calls with the same character set as used by the region-specific command line interpreter "cmd.exe".

For example, "Codepage 850", also known as "DOS-Latin-1", applies to the region Western Europe.

The "Notepad" editor integrated in Windows does not support saving files with this character set.

Note

Uppercase/lowercase

For the command line switches and arguments, **always** use the specified uppercase/lowercase syntax.

Note

Absolute/relative path

You can specify the absolute path or the relative path.

The relative path with `".\"` or `"./"` refers to the current work folder, where the batch or CMD was called.

If no path is specified, the work folder is used.

Parameter

Significance of the switches

Exporting a step as an external step project
`Usage: exportstep [options] <projectfile>`
Arguments:

Exporting a step as an external step project	
`projectfile Expert project (.upproj)`	
Options:	
`-h|--help`	Display Help
`-s|--stepid`	The ID of the step to be exported
`-o|--output`	File "*.upstep" of the exported project with complete path
`-c|--cleantarget`	If the target folder is not empty, clear it first. Only empty folders or a folder with an existing step project is allowed. Folders containing other files are not emptied, and the step project is not exported.
`-x|--xmllog`	Write XML report

Return values

Table 8-21 Return values Expert step project

Value		**Description**
0	NO_ERRORS_OR_WARNINGS	No errors and no warnings occurred
3	INVALID_ARGUMENTS	Incorrect parameter specified
32	NO_PROJECTFILE	No project file available
33	INVALID_PROJECT	Invalid project
34	INVALID_ANSWER_MODE	Invalid answer mode
35	NO_DESTINATION_FILE_SPECIFIED	No destination file specified
36	INVALID_PROJECT_SCHEMA	Invalid project schematic
37	UNSUPPORTED_PROJECT_VERSION	Unsupported project version

Value		Description
4	FAILURE	Runtime error
43	TARGET_DIRECTORY_NOT_EMPTY	Target directory is not empty

Example

```
@echo off

"C:\Program Files\SIEMENS\Automation\SINUMERIK\Create MyConfig 6.3\ExpertCmd.exe"
exportstep -s 2 -o D:\td\upstep\part2.upstep -c d:\td\master3\master3.upproj

if %ERRORLEVEL% EQU 43 (
        echo "Directory could not be emptied!"
        pause
)

if NOT %ERRORLEVEL% EQU 0 (
        echo "Error when exporting step project"
        pause
)

if %ERRORLEVEL% EQU 0 (
        echo "Exported without any errors"
        pause
)
```

8.5 Managing and using configurations

8.5.1 Overview

All settings you define in the property fields under "Configuration", "Dialogs" and "Steps" are configurations.

Some of the configurations are mapped to system variables and can be written or read subsequently via scripts, e.g. during or before execution.

When the term "configurations" is used within this section, just this subset of configurations for which system variables also exist is meant.

In this case, the handling of configurations always means the handling of the associated system variables, including their values.

The information about which system variables are available for "Configuration", "Dialogs" and "Steps" can be found in the "Home area" by calling up the context-dependent button for system variables.

Note

Reference list, system variables

More information on the individual system variables can be found in section "Reference list system variables (Page 380)".

The configurations are located in text files with extension "*.upcfg".

The format of the text files corresponds to that of a CMC task or a CMC script.

Value assignments are made in the form <`System variable`> = <Value>. Comments are identified by a prefixed semicolon. You can also remove any number of configurations from the configuration files and, for example, so limit the number being used. When loading/using this configuration, only the settings contained in the package are set with all other configurations remaining unaffected. Proceed in this manner when, for example, in an existing package, a switchover is performed only between a new installation and an upgrade for automatic execution.

8.6 Actions

In order for an action to be executed when the package is run, an action must be linked to a step and the step must be activated for execution An existing action script can permit or prevent execution of the action.

An action is configured for example by creating instructions in the action script, new tasks created and processed and required tasks and/or files linked.

Every action is opened in a separate tab. The tab also includes an action script, which you can populate with your instructions. Folders and new tasks can also be easily created and new files linked.

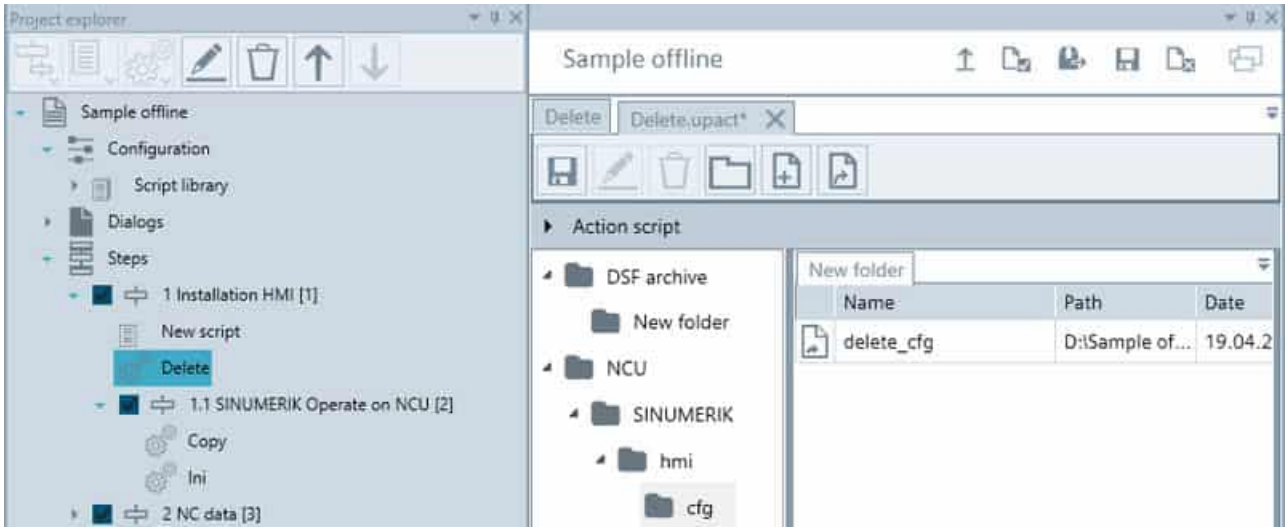


Figure 8-64 Example - "Delete" action tab

Creating an action

An action is always created below a step, and is therefore automatically assigned to this step.

Proceed as follows to create an action:

1. Select the step for which you wish to create an action.
2. Click on the required action in the toolbar above the "Add action" button. The action is created, and linked with the step.

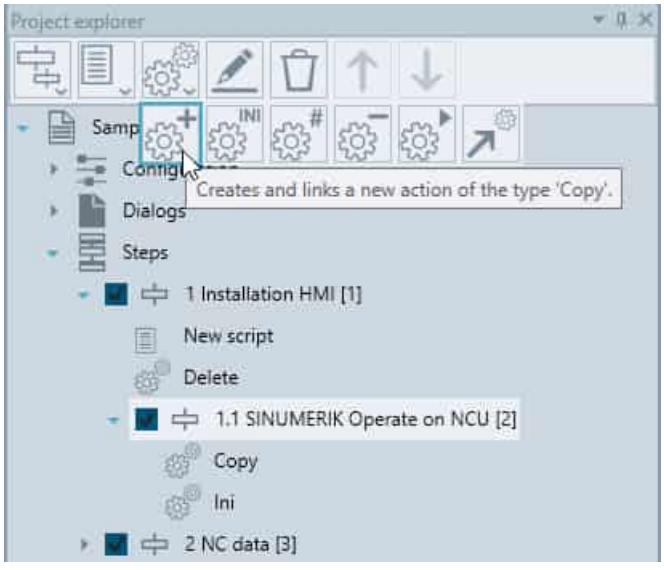


Figure 8-65 Creating an action

Action types

You can create the following actions:

- Copy (Page 361)
- Change INI (Page 364)
- Create with preprocessor (Page 368)
- Delete (Page 373)
- Execute (Page 375)
- Link an existing action

Table 8-22 Action overview

Actions	Supported target areas	Function
Copy (Page 361)	• DSF archive • NCU • PCU	Copies folders and files in the target area
Change INI (Page 364)		Allows text files in INI format to be adapted
Create with preprocessor (Page 368)		Enables the script-controlled creation of text files
Delete (Page 373)		Deletes files and folders in the target area
Execute (Page 375)	• NCU • PCU • Windows (runtime system) • Linux (runtime system)	Allows a program, command or script to be executed

Linking an Action

All links are only resolved when the package has been deployed and the data has been accepted in the package. The advantage of the link is that the source data can be refreshed or edited independently of the project before deployment.

Action in package execution

For an Action to be executed when the package is executed, the step linked to the Action must be activated using a checkmark in the check box.

During package execution, the actions are executed from top to bottom in the sequence of the listing in the step tree depending on the steps. An action is activated when the step linked to the action is activated.

For execution of an action, it makes no difference whether this action is open or closed.

All the relevant instructions and data are inserted in the respective action under the relevant tab during the configuration.

You can save an action script for every action, which, for example, controls the execution of the action.

Note

Sequence and execution of actions in the step tree

Only the sequence of the steps in the step tree is relevant for the sequence and execution of the actions. The actions are assigned to the steps respectively. Actions are processed in the order below a step.

Sequence when executing actions

An action is always linked with a step. As a consequence, always change actions in the sequence of their execution below a step.

You change the sequence of actions below a step as follows:

1. Select the required action in the project tree
2. In the toolbar shift the action either up or down using the arrow keys

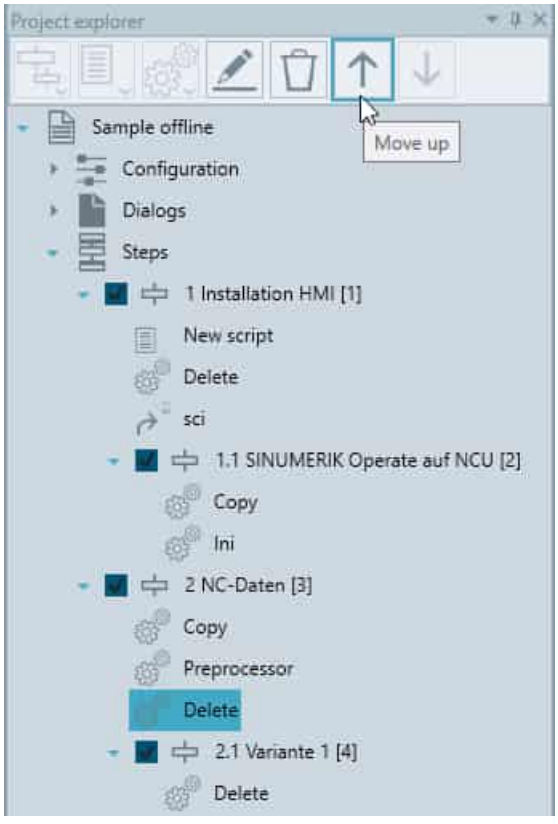


Figure 8-66 Changing the sequence of actions

Editing an Action

Double click an action in the step tree to edit it. The Action is displayed as in a tab.

"Action scripts" and linked files of an action are directly edited in the integrated "Script Editor". This has the advantage that you always obtain support from the Expert.

8.6.1 Action properties

You can configure the following property group for each action:

- Action

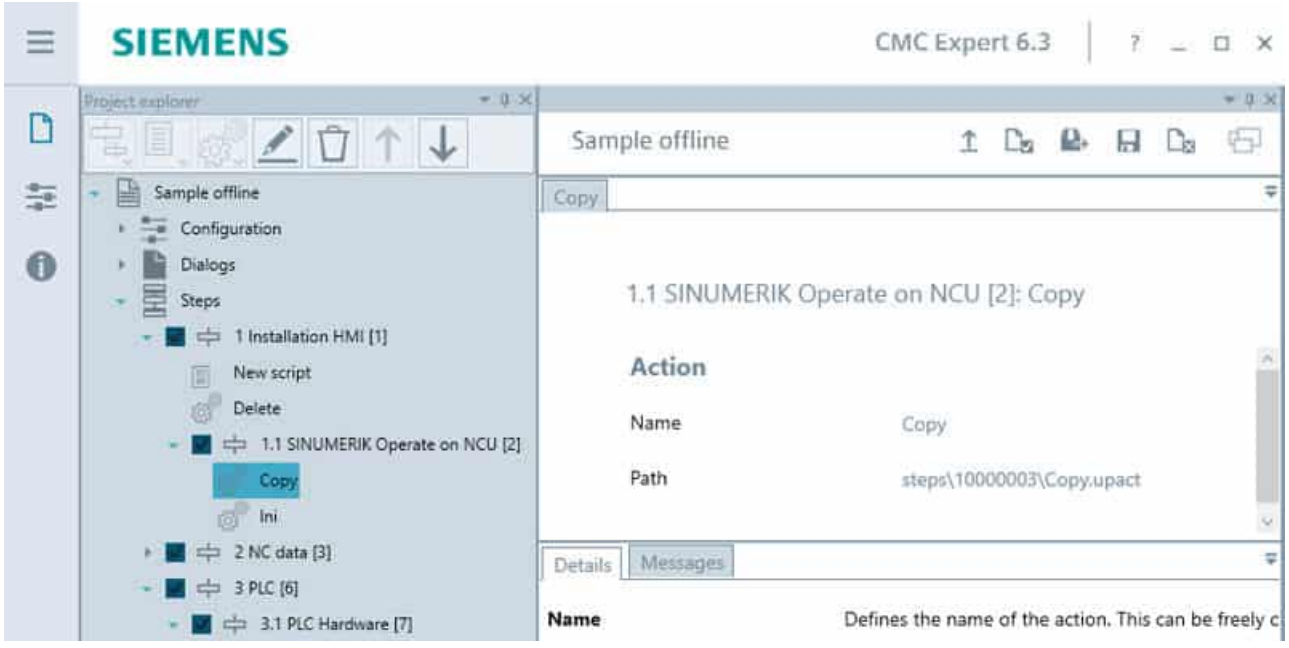


Figure 8-67 Copy properties

Action

The following settings are available for the "Action" property group:

Name	• Copy • Ini • Preprocessor • Delete • Execute	Display of file name As default, the action file is named according to the type of action. You can rename an action in the step tree using the appropriate toolbar button.
Path	steps\10000008\Copy.upact	Displays the path of the linked file. Based on the path, you can assign the action to a step by specifying the path ID of the step. The path ID of the step is automatically assigned and cannot be changed.

8.6.2 Action scripts

You create an "action" below a step. The "Action script" is part of the "Action", and is therefore automatically stored for each action. In the open tab of an action, edit the "action script" in the integrated "Script Editor" of Expert.

Executing action scripts

Executing actions and the existing action scripts is shown schematically in the following diagram.

The procedures, which could potentially control the sequence of the actions or action scripts, are shown as example in the following diagram.

In principle, you can use all script language elements in the "action script".

More information on script languages is provided in Section "Create MyConfig CMC script", taking into consideration the constraints in Section "Data access (Page 50)".

The shortcut menu supports you when creating an action script.

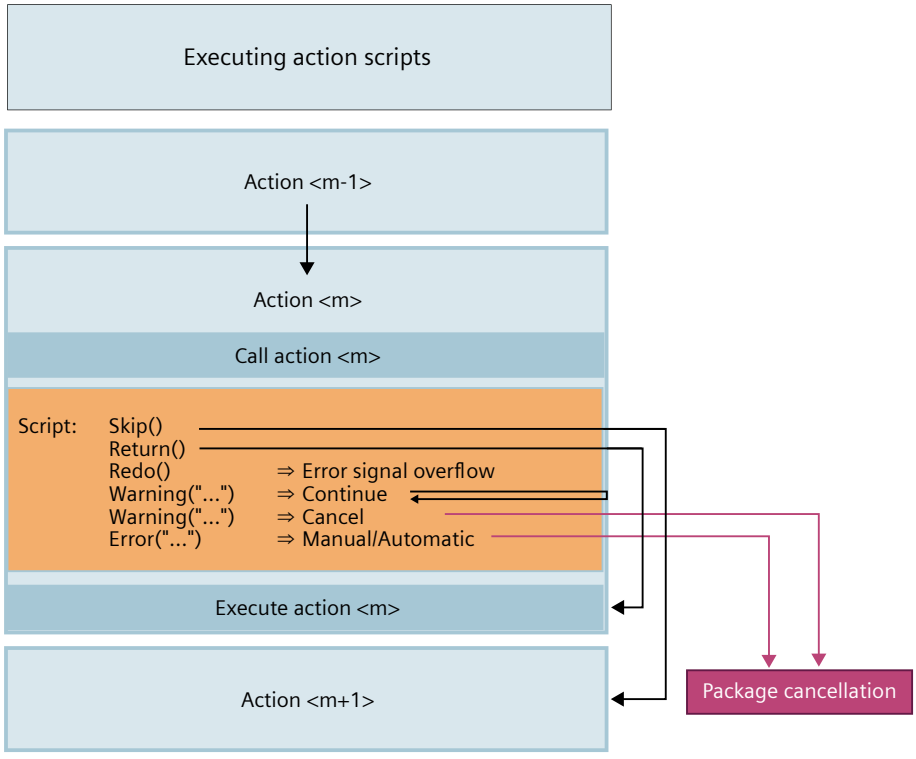


Figure 8-68 Execute actions

Opening and editing an action script

Every "action" already includes an "action script".

To open and edit an "action script":

1. Double-click on the required action below the "Project explorer" in the step tree.

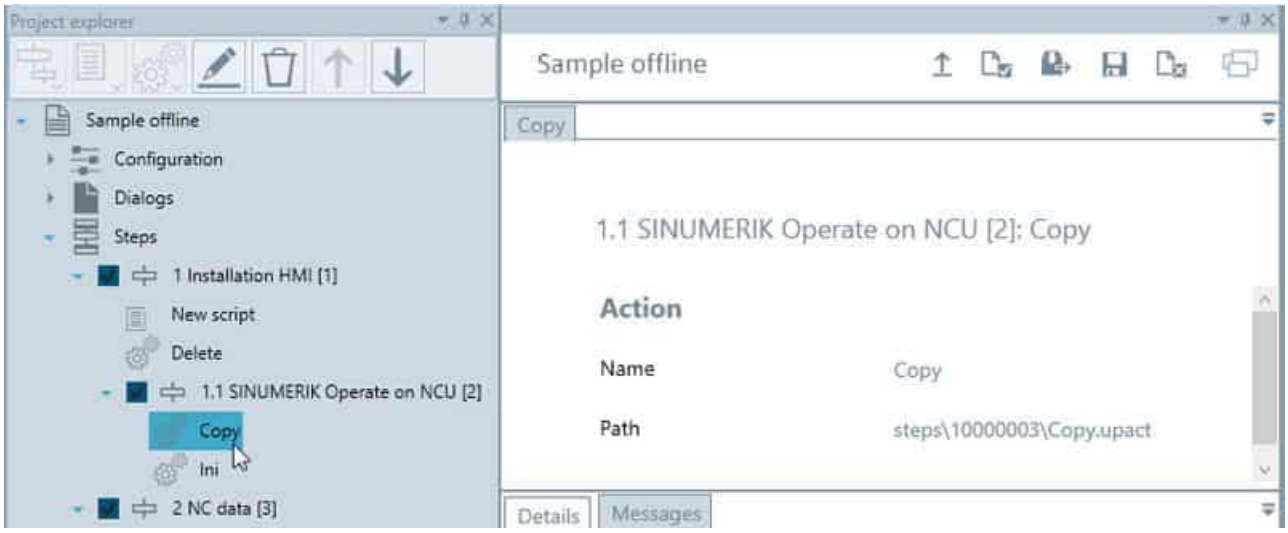


Figure 8-69 Double-click action

2. The action opens as tab with the integrated "Action script"; here, using a "Copy" type action with "Copy orders" as example.

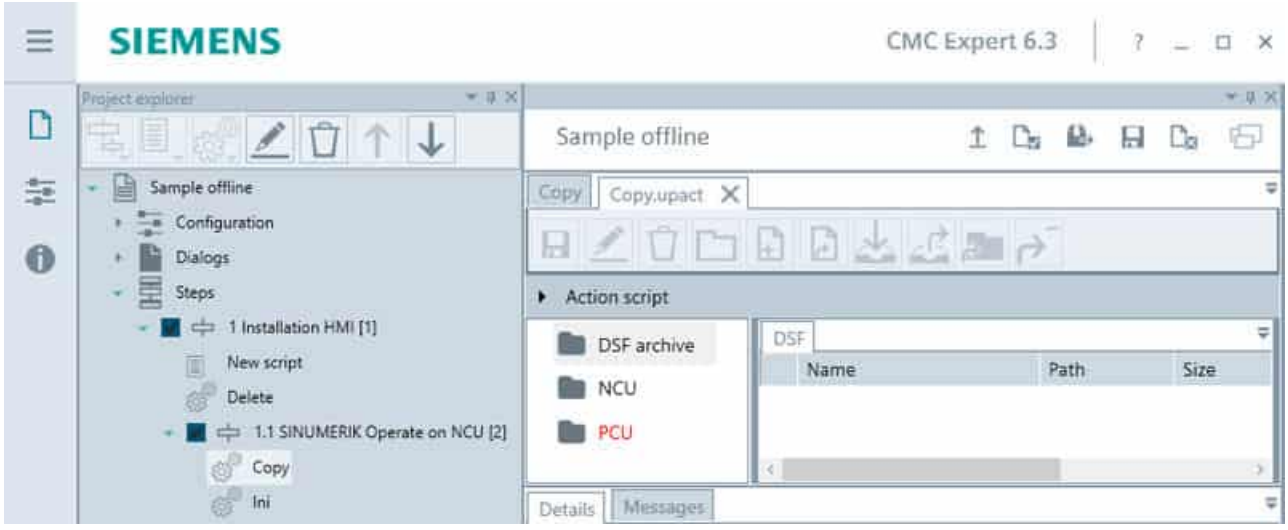


Figure 8-70 Opening an action tab in the Home area

3. To edit the action script, click on the "Action script" title bar in the Home area.
   Enter your instructions in the action script. You have the option of working with the shortcut menu, and also inserting scripts, for example.

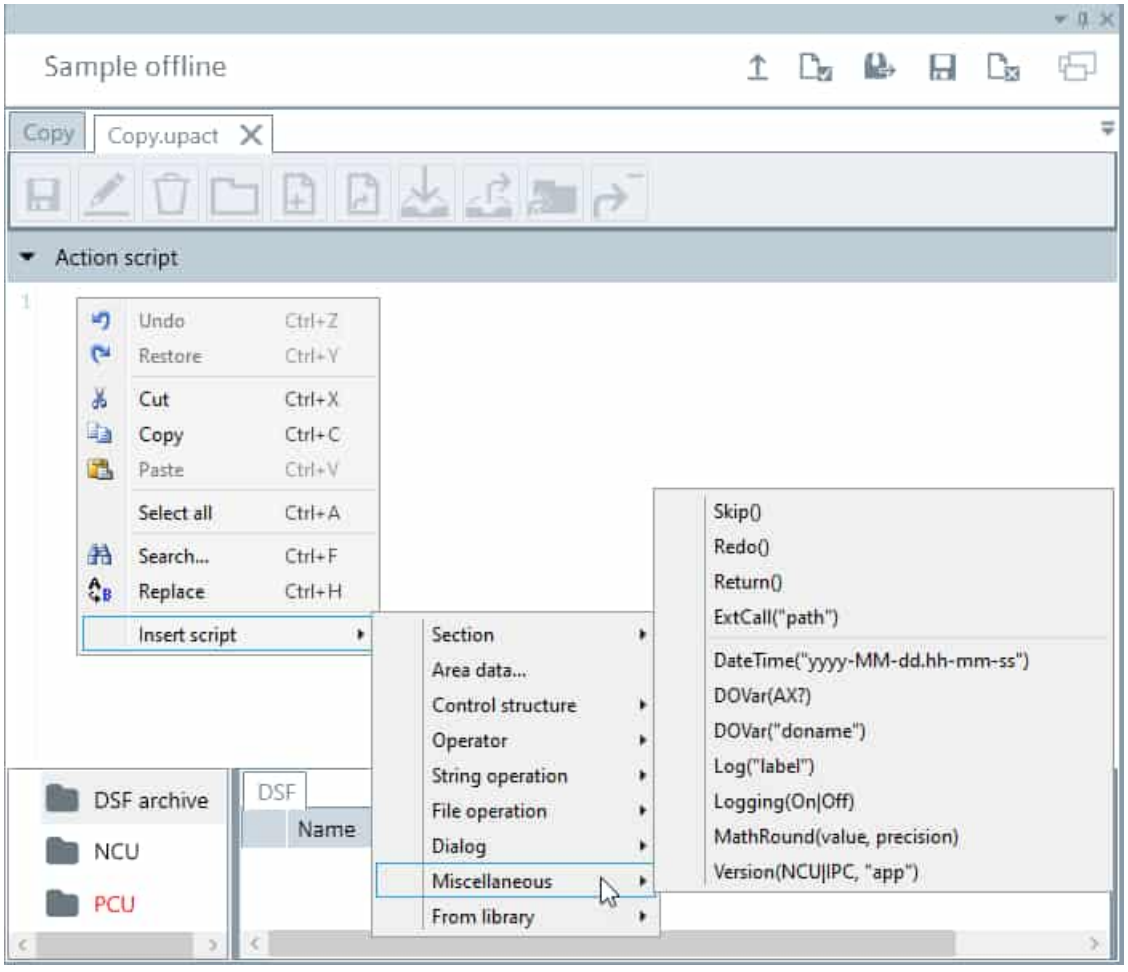


Figure 8-71 Opening and editing an action script

Shortcut menu action script

The following shortcut menu is available for action scripts:

Table 8-23 Action script - shortcut menu

	Undo	Ctrl + Z
	Restore	Ctrl + Y
	Cut	Ctrl + X
	Copy	Ctrl + C
	Paste	Ctrl + V

	Select All	Ctrl + A	
	Find...	Ctrl + F	
	Replace...	Ctrl + H	
	Insert script		Section (Page 53)
			Area data...
			Control structure (Page 54)
			Operator (Page 57)
			String operation (Page 62)
			File operation (Page 65)
			Dialog (Page 77)
			Miscellaneous (Page 97)
			From library (Page 103)

More information on the script language is provided in the section or the online help "Create MyConfig - CMC script (Page 29)".

See also

Script language (Page 380)

8.6.3 Copy

8.6.3.1 Description of the "Copy" action

Action of the type "Copy"

The "Copy" action type copies folders and files in the target area.

Note

Renaming linked folders

You can subsequently rename linked folders using the toolbar button.

Renaming is necessary if the folder on the control is to have a different name than in your database to which the folder is linked.

Note

Selected data areas define target areas

Possible target areas are automatically displayed in the "Copy.upact" tab in the target directory.

Whether a target area can be used depends on the data areas (archive, NCU/PPU, PCU) that were activated in the configuration.

The data areas are defined in the "Project explorer" in the Configuration (Page 236) under "Data areas".

Note

Display of the target area

Red - target area is not processed/executed

If a data area is not configured or is subsequently deactivated, then the target area node is shown in red.

The target area (e.g. DSF archive, NCU) cannot be processed/executed as the configuration is missing.

Target areas

The following target areas are supported:

Table 8-24 Target areas, "Copy" action

Target area	Description
DSF archive	Archive image
NCU	NCU/PPU root
PCU	PCU drives/shares

Create copy orders

In the target areas, new folders and new files can be created or linked using the shortcut menu. You can enter new files or folders as link using the toolbar button. Links are deleted again via the toolbar.

8.6.3.2 Linking the directory structure

Using the "Copy" action, complete directory structures and data content can also be transferred into a target area.

Procedure

Link a target directory in the target area with the folder in which you saved your data.

1. Open action "Copy.upact" in the home area.
2. In the home area, navigate to the empty folder in which you wish to link a directory structure with the data content.

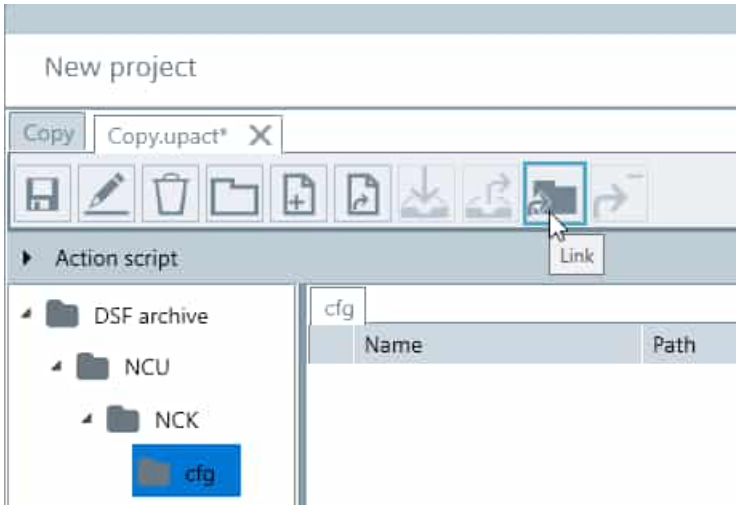


Figure 8-72 Linking the folder structure

3. Navigate to the externally stored directory structure and select the folder to be linked. Confirm with "Open".

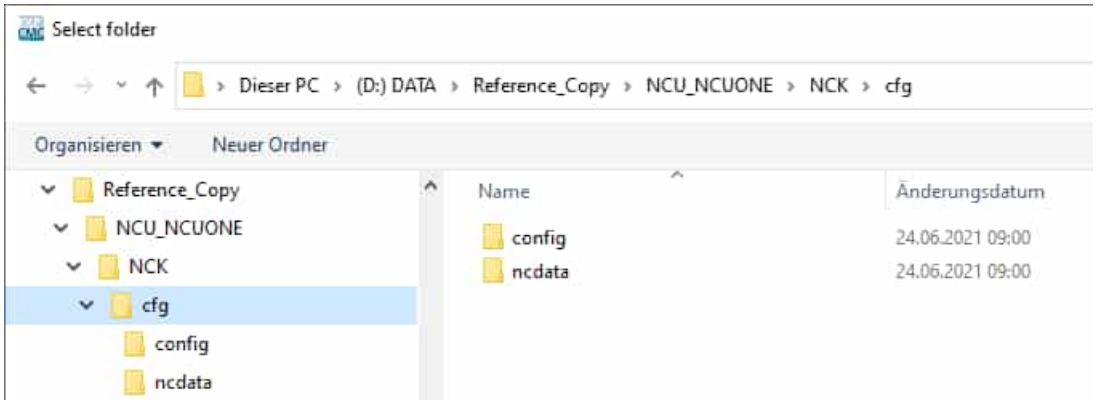


Figure 8-73 Selecting a folder

4. The directory structure with all of the data content is linked.
   The folder names shown in blue indicate that the directory structure has been linked.

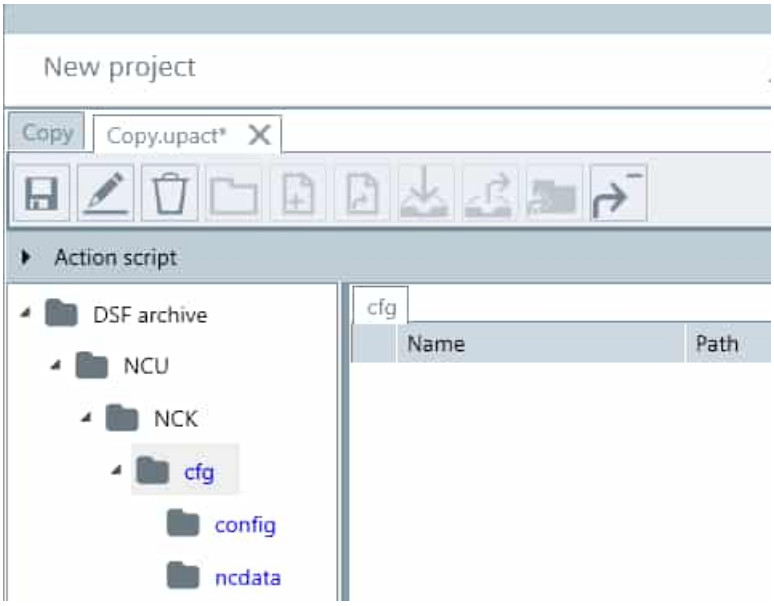


Figure 8-74 Folder structure linked

The link means that the structure under folder "cfg", as well as all files in folder "cfg" and in the subdirectories are copied.
The advantage between linking individual data and this type of link is that you link complete structures with just one operation.
The link can also be deleted.

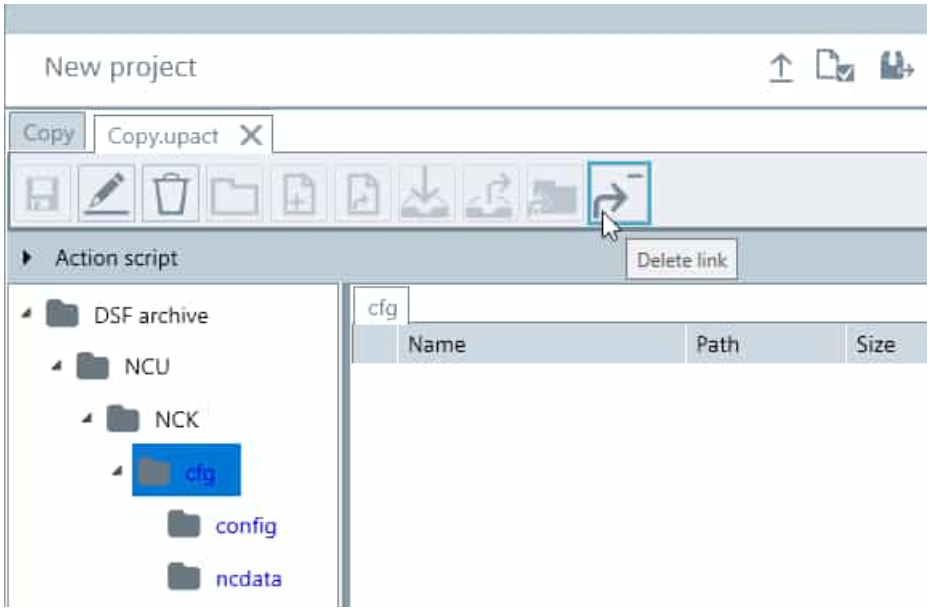


Figure 8-75 Deleting a link

8.6.3.3 Notes on the "Copy" action

Note

Case sensitivity on the NCU/PPU

In contrast to Windows, the file system used on the NCU/PPU differentiates between upper case and lower case letters.

8.6.4 Change INI

8.6.4.1 Description of the "Change INI" action

Action of the type "Change INI"

The "Change INI" action allows you to adapt text files in the selected target area.

This action is specifically available for INI files that comprise one or more sections and value assignments included in them:

```
[Section]
Identifier=value
```

Note

Selected data areas define target areas

Possible target areas are automatically displayed in the "Ini.upact" tab in the target directory.

Whether a target area can be used depends on the data areas (Archive, NCU/PPU) that were activated in the configuration.

The data areas are defined in the "Project explorer" in the Configuration (Page 236) under "Data areas".

Note

Display of the target area

Red - target area is not processed/executed

If a data area is not configured or is subsequently deactivated, then the target area node is shown in red.

The target area (e.g. DSF archive, NCU) cannot be processed/executed as the configuration is missing.

Target areas

The following target areas are supported:

Table 8-25 Target areas for "Change INI" action

Target area	Description	Example
DSF archive	Archive image	Predefined folder names are available in the shortcut menu. Folders can also be created additionally as required via the shortcut menu or toolbar.
NCU	SD card of the NCU/PPU Task: • INI file	
PCU	PCU drives/shares	C:, D:

Creating change orders

To create a change task, in the project Explorer, double click on a "Change INI" action. In the Home area, a "Ini.upact" tab with integrated "action script" opens.

The directory structure with the target area nodes (DSF archive, NCU) is located below the action script on the left side.

On the right side, the respective selected tab is displayed, where you can create a new task or link a new file.

You have the following options to create change orders:

- **New task**
  - You write change instructions directly into the action script. The task name must be identical to the file to be changed.
  - To create a new task, select a target area under "Change orders":
    Create the new task using the appropriate toolbar button. The task is automatically linked with the previously selected target area.
    The task is opened with a double-click, and is edited in the integrated "Script Editor".
- **Link task**
  - A file is linked using the toolbar button: "Link new file"
    The linked file can be opened and edited using the integrated "Script Editor".

Note

Using a third-party editor for CMC

A third-party editor can be used for CMC.

Make sure that you use ASCII encoding or UTF-8 encoding (without BOM) when saving the file.

Use UTF-8 encoding as soon as you use umlauts and special characters outside the ASCII 7-bit character set in the code characters.

On deployment, the change task is accepted in the package.

Script Editor

Based on the structure of the configuration files, enter change instructions, specifying the section.

Change instructions comprise identifier, operator and possibly the value. Enter this information line by line. The shortcut menu supports you when inserting operators.

Lines that start with a semicolon `";"` are not evaluated and can be used for comments.

"Special operators (Page 61)" can be inserted anywhere in a line to insert the contents of Up variables for "Modify".

Note

If folders, files, sections or identifiers are not available, then the package itself generates these to implement a value assignment.

Operators

=	Assignment of a new value
~	Delete date: Identifier and value are removed
?=	Assigning a (new) value if no value is available

Example:

```
[LANGUAGE]
FontList=Europe, Europe, Europe, Europe, Europe
[Local]
ADRESS0=0,NAME=/NC,LINE=11
ADRESS1~
[Function]
Del. Dir query1?=111
```

8.6.4.2 Example action "Change INI"

Example

The "ini" extension is generally used for configuration files. You can change all files that comply with the structure of an INI file.

For the package execution in the "INI file" mode, in the change task, the specified file is opened via the directory path, and the operations specified in the relevant section executed.

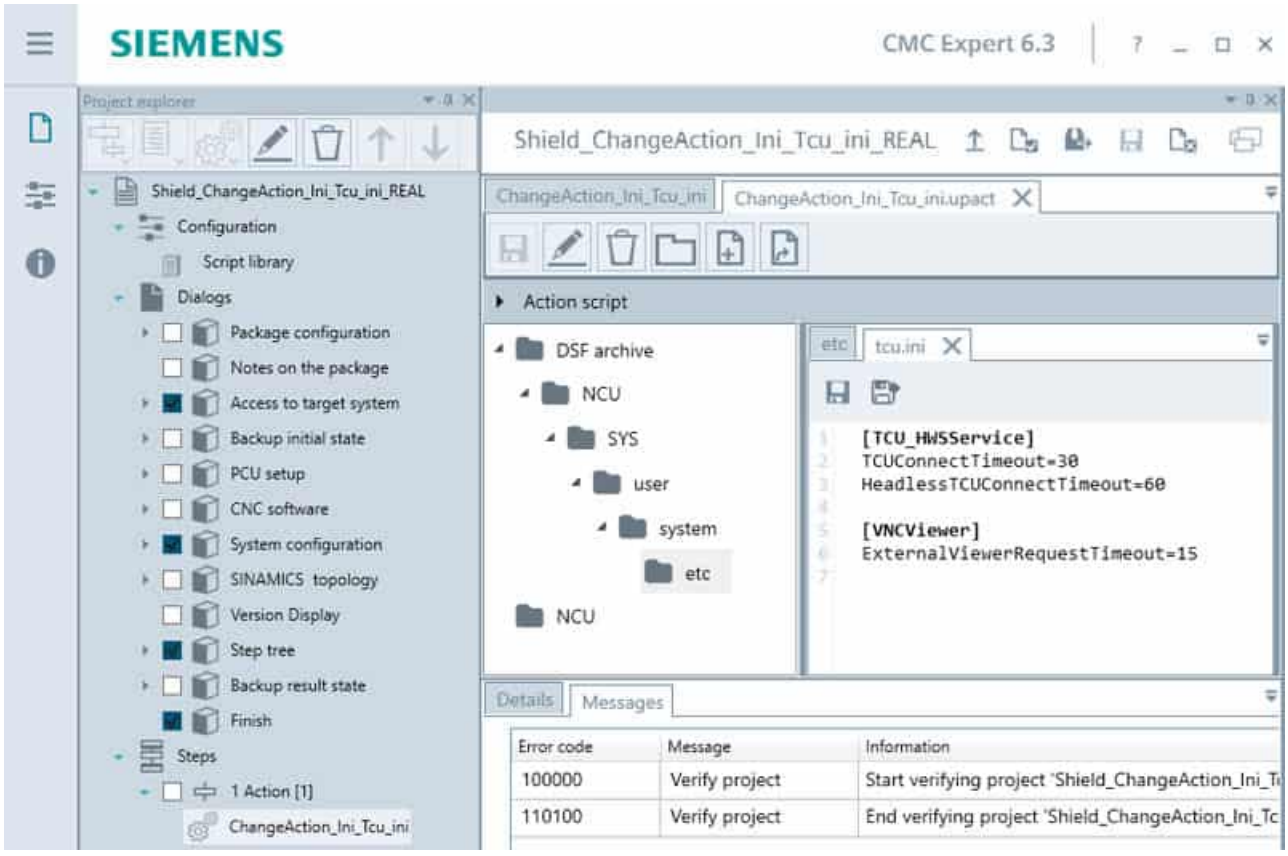


Figure 8-76 Example INI file

In the example, the INI file in the device is replaced by the change task and therefore the existing values replaced by the new values from the INI file.

If the file or section does not exist in the file, the package in the target system automatically creates the file or section in the existing file.

8.6.5 Create with preprocessor

8.6.5.1 Description of the "Create with preprocessor" action

Action of the type "Create with preprocessor"

The "Create with preprocessor" action allows you to adapt text files in the selected target area.

Note

Selected data areas define target areas

Possible target areas are automatically displayed in the "Preprocessor.upact" tab in the target directory.

Whether a target area can be used depends on the data areas (archive, NCU/PPU, PCU) that were activated in the configuration.

The data areas are defined in the "Project explorer" in the Configuration (Page 236) under "Data areas".

Note

Display of the target area

Red - target area is not processed/executed

If a data area is not configured or is subsequently deactivated, then the target area node is shown in red.

The target area (e.g. DSF archive, NCU) cannot be processed/executed as the configuration is missing.

Target areas

The following target areas are supported:

Table 8-26 "Create with preprocessor" target areas

Target area	Description	Example
DSF archive	Archive image	When required, folders can be created using the shortcut menu or toolbar.
NCU	SD card of the NCU/PPU Task: • Preprocessor script	
PCU	PCU drives/shares	C:, D:

Creating change orders

To create a change request, double-click a "Create with preprocessor" action in the Project explorer. In the Home area, a "Preprocessor.upact" tab with integrated "action script" opens.

The target area nodes (DSF archive, NCU) are located below the action script on the left side. After selecting a target area node, predefined folders or new folders can be created via the shortcut menu.

A new task is created via the button of the same name in the toolbar.

A new file is also linked by clicking the corresponding button in the toolbar.

On the right side, the respective selected tab is displayed with the available linked tasks or files. Tasks and files are opened with a double click and can be edited using the integrated script editor.

The following options are available for creating change orders:

- **New task**
  - You can write change instructions directly into the action script. The task name must be identical to the file to be changed.
  - To create a new task, select a target area node:
    Create the new task using the appropriate toolbar button. The task is automatically linked with the previously selected target area.
    The task is opened with a double-click, and is edited in the integrated "Script Editor".
- **Link file**
  - A file is linked using the toolbar button: "Link new file"
    You can both create and link a new file or link an existing file.
    The linked file can be opened and edited using the integrated "Script Editor".

Note

Using a third-party editor for CMC

A third-party editor can be used for CMC.

Make sure that you use ASCII encoding or UTF-8 encoding (without BOM) when saving the file.

Use UTF-8 encoding as soon as you use umlauts and special characters outside the ASCII 7-bit character set in the code characters.

On deployment, the change task is accepted in the package.

See also

Special operators (Page 61)

8.6.5.2 Examples of "Create with preprocessor" action

Example 1

In the example, a "`Cycle1.spf`" cycle is created in the "`/NCU/NCK/cycles/cma.dir`" path in the archive.

Using an example, it is shown that you can use "Pragma Eol" for the Windows system (CRLF lines 3 - 11) as well as for the LINUX system (LF lines 14 - 29).

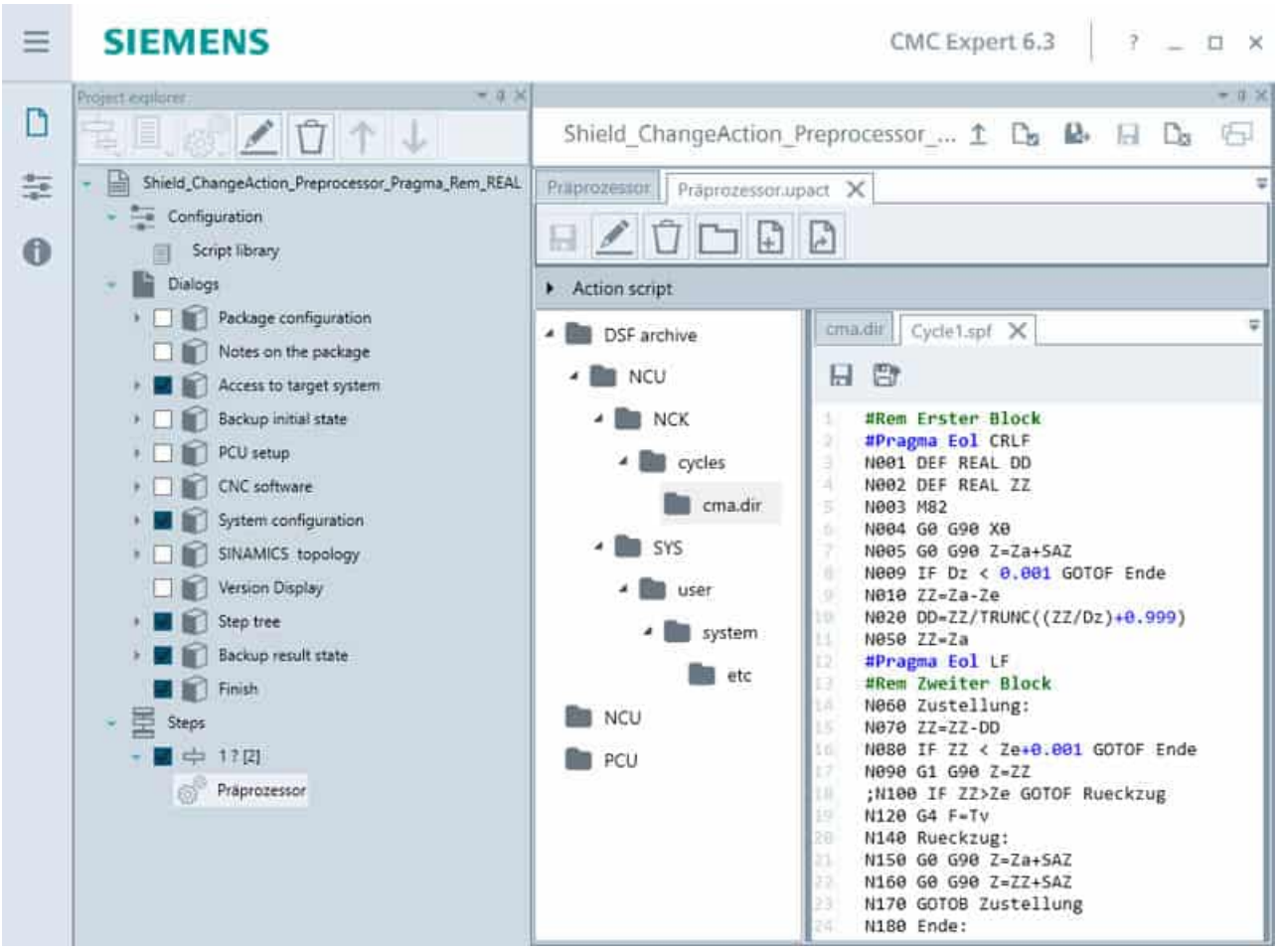


Figure 8-77 Example 1 - "Create with preprocessor" action

Example 2

In the "Create with preprocessor" action, contents can be incorporated script-controlled in files.

Edit the file specified in the directory structure in the corresponding folder. The contents of the file are always newly created script-controlled; existing content is deleted.

If the file does not exist, the package automatically creates the file in the target system.

Preprocessor instructions always begin with "`#`". Only these lines are evaluated as script by the CMC package. Lines that do not begin with this character are imported into the file to be created without checking.

The following preprocessor instructions are valid:

`#Pragma Eol CRLF`	Defines the "end of line indicator" for the file as "`CRLF`".
`#Pragma Eol LF`	Defines the "end of line indicator" for the file as "`LF`".

`#Rem`	Marks a line as comment.
`#If`, `#ELIF`, `#ELSE` and `#ENDIF`	For controlled creation. Instead of `#ELIF`, you can also use `#ELSIF`.
`##`	For inserting lines that begin with `#`. For creating preprocessor scripts by a preprocessor script.

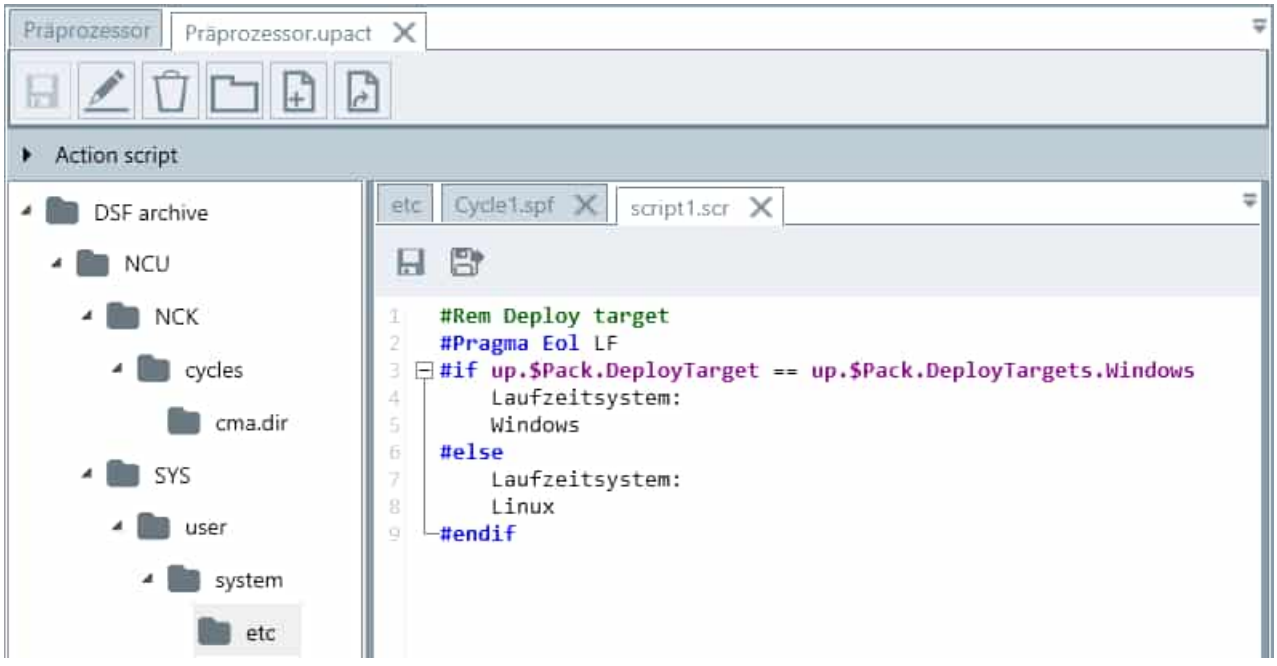


Figure 8-78 Example 2 - "Create with preprocessor" action

8.6.6 Delete

8.6.6.1 Description of the "Delete" action

Action of the type "Delete"

The "Delete" action deletes files and folders in the target area.

Note

Selected data areas define target areas

Possible target areas are automatically displayed in the "Delete.upact" tab on the left in the target directory.

Whether a target area can be used depends on the data areas (archive, NCU/PPU, PCU) that were activated in the configuration.

The data areas are defined in the "Project explorer" in the "Configuration (Page 236)" under "Data areas".

Note

Display of the target area

Red - target area is not processed/executed

If a data area is not configured or is subsequently deactivated, then the target area node is shown in red.

The target area (e.g. DSF archive, NCU) is not processed/executed as the configuration is missing.

Target areas

The following target areas are supported:

Table 8-27 "Delete" action target areas

Target area	Description	Example
DSF archive	Archive image	When required, you can create folders using the shortcut menu or toolbar.
NCU	NCU/PPU root	
PCU	PCU drives/shares	C:, D:

Creating delete instructions

To create a delete instruction, in the Project explorer, double click on the required "Delete" action. A tab with the default name of the delete action "Delete.upact" opens in the home area, which already contains an "action script" and a directory with the target areas (e.g. DSF archive, NCU). You can change the name of the action in the Project explorer using the "Rename" button.

You create a new task using the toolbar button with the same name.

If required, you can divide the delete instructions into several tasks to improve the structure.

You can freely select task names.

When executing, the files specified in the tasks are deleted without any prior confirmation prompt.

You can create task files in the following ways:

- New file
  - To create a new task file, select and mark a target area in the directory.
    Create a new task file using the appropriate toolbar button. Then, specify the name of the task file.
    The file is automatically linked with the target area.
    You can then open and edit the file by double-clicking in the integrated "Script Editor"of Expert.
- Link new file
  - An external file that contains instructions is linked as a task file.
    First select the target area, and then the task file to be linked using the toolbar button "Link new file".
    The task name is allocated the name of the linked file. You can subsequently rename file names using the toolbar button "Rename".
    You can edit the linked file, either externally or in the "Script Editor"; the supporting shortcut menu Expert is available in the "Script Editor".
    On deployment, the task is accepted in the package.

Notes regarding file and folder names

The names of the files and folders to be deleted are specified in the "Delete instructions".

The following wildcards are permitted in the file names:

- "*" - for none or one or several arbitrary characters
- "?" - one arbitrary character

On the other hand, no place holders are permitted in the folder names. Folders are deleted with their complete content.

Only one delete operation is permissible for each line.

If a line starts with a comment character ";", then the line is not evaluated.

Examples:

`Cycle*.spf`	All cycles in the current folder are deleted.
`*.*`	All files in the current folder are deleted; subfolders are kept.
`4711.WPD`	Folder 4711.WPD with all of its contents is deleted.

8.6.6.2 Notes regarding the delete action

Note

Case sensitivity on the NCU/PPU

In contrast to Windows, the file system used on the NCU/PPU differentiates between upper case and lower case letters.

8.6.7 Execute

8.6.7.1 Description of "Execute" action

Execute action

The "Execute" action allows a program, command or script to be executed. For example, you can install software components from SIEMENS or other suppliers - or perform special manipulations in the archive or other target areas.

Note

Selected data areas define target areas

Possible target areas are automatically displayed in the "Execute.upact" tab in the target directory.

Whether a target area can be used depends on the data areas (archive, NCU/PPU, PCU) that were activated in the configuration.

The data areas are defined in the "Project explorer" in the "Configuration (Page 236)" under "Data areas".

Note

Display of the target area

Red - target area is not processed/executed

If a data area is not configured or is subsequently deactivated, then the target area node is shown in red.

The target area (e.g. DSF archive, NCU) is not processed/executed as the configuration is missing.

Note

Display of the runtime system Windows / Linux

Red - the runtime system is not used

The runtime system is selected in the "Configuration" under "Deployment":

- Windows package (PC/IPC)
- Linux package (NCU)

The runtime system that is not configured, is marked in red in the "Execute.upact" directory. Files can be allocated; however, they are not executed. The check run issues a warning.

Target areas

The following target areas are supported:

Table 8-28 Target areas "Execute" action

Target area	Description	Example
NCU	Root of the NCU/PPU	-
Windows	Runtime system	-
Linux		-

Execute, general

Enter all the files and folders required for "Execute" as link under the appropriate target area. When deployed, the links are resolved and the folders and files are accepted in the package.

You can specify a command or an executable file in the "Execution of:" text field. You can also pass arguments separated by spaces, e.g. file names, switches.

Write commands or file names that contain spaces in straight quotation marks.

Special operators (Page 61) can be inserted anywhere in the text field.

Execute - NCU

During execution, all folders and files are unzipped into a temporary folder. This temporary folder is used as work directory when "executing" and is then deleted.

Note

Temporary folder

The temporary folder is not the package storage. This means a "`./`" used for "Execution of" does not point next to the package, but to the temporary folder.

The script outputs are transferred to the logbook. In this case, the error messages (stderr) of the programs called in the script are entered in *italics*.

The return value of the script is evaluated as terminating error code. A return code not equal to "0" results in an error message. The user must decide whether the package should be canceled or continued.

More information can be found in Section: "Notes relating to the Execute action (Page 379)"

Execute - Runtime system

A package can be deployed for various runtime systems.

Two scripts must be written to deploy a package for Linux and for Windows that makes special changes via "Execute" in the "DSF archive" area.

Depending on the runtime system, Windows or Linux, the corresponding node is executed automatically and the other skipped during execution.

All folders and files are unzipped to a temporary folder; this folder is then used as the work directory and subsequently deleted.

Note

Temporary folder

The temporary folder is not the package storage. This means a "`./`" used for "Execution of" does not point next to the package, but to the temporary folder.

The script outputs are transferred to the logbook. In this case, the error messages (stderr) of the programs called in the script are entered in *italics*.

The return value of the script is evaluated as terminating error code. A return code not equal to "0" results in an error message. The user must decide whether the package should be canceled or continued.

Example

All files, except for "`OEM001.SPF`", should be deleted from the `CMA.DIR` directory of the NC.

As there is no special delete instruction for this purpose, a script is generated which saves the file to be obtained, subsequently deletes the folder and then returns the file.

The particular script file is copied to the component and entered as a command for "Execute".

Windows

```
Md .\Save
Move %UP_ARC%\NC\NCU_<name>\NCK\cycles\cma.dir
\OEM001.SPF .\Save\
Del /Q %UP_ARC%\NC\NCU_<name>\NCK\cycles\cma.dir\*.*
Move .\Save\OEM001.SPF %UP_ARC%\NC\NCU_<name>\NCK\cycles
\cma.dir\
```

Linux

```
mkdir ./Save
mv $UP_ARC/NC/NCU_<name>/NCK/cycles/cma.dir/OEM001.SPF ./
Save/
rm -r $UP_ARC/NC/NCU_<name>/NCK/cycles/cma.dir/*
mv ./Save/OEM001.SPF $UP_ARC/NC/NCU_<name>/NCK/cycles/
cma.dir/
```

8.6.7.2 "Notes relating to the Execute action"

Note

System variables for the environment

Use "System variables for the environment" in a script file for Windows "cmd" or Linux "sh", because only these script files are known to the script interpreter.

If, however, "System variables for the environment" are used in "Execution of" the component, for Linux use a "sh" as prefix, and for Windows, a "cmd" as prefix before the command.

The "System variables for the environment" can be used in all CMC scripts.

Example:

```
If !FileExist(RTS,"%UP_PACK%/WCD/file.txt")
```

As alternative to `UP_PACK`, the system variable `Up.$Pack.Dir` can also be used for "Execution of" and in all CMC scripts.

Example:

```
md "$(Up.$Pack.Dir)/WCD"
If !FileExist(RTS,"$(Up.$Pack.Dir)/WCD/file.txt")
```

System variables for the environment

`UP_PACK` - Location from where the package was started

`UP_ARC` - Temporary storage location of the archive tree with the NC, PLC, CP and DRV subfolders

`UP_CFID` - ID of the SD card

`UP_PCU_ROOT` - Computer name of the "PCU".

Available only for packages with the "PCU" data area.

`UP_ARC_ROOT` - Folder of the archives to be processed.

Available only for packages that use only the "Use archive" data area.

Note

Exit Script

If no script-internal error handling is performed, a script should preferably always be exited with exit "0".

Note

Command interpreter under Windows

For Windows, prefix the command line scripts to the command interpreter.

For example:

```
Cmd.exe /c test.cmd
```

Note

Using shell scripts for Linux

When using shell scripts "`*.sh`" for Linux, always ensure that in the Unix file format you only save LF; and not CR/LF.

The shell identifier must be located in the first line, e.g.:

```
#!/bin/sh
```

The script must be started using a shell call, e.g.:

```
sh my_script.sh
```

If you specify commands directly with place holders or environment variables, then use a shell as prefix e.g.:

```
sh rm /user/*.log.
```

If you specify a file for execution, then use "`./`" as prefix, e.g.:

```
./test.sh.
```

8.7 Script language

More information about the script language is provided in Section "Create MyConfig CMC script (Page 29)" or in the online help with the same name.

8.8 Reference list, system variables

8.8.1 General information about system variables

System variables for Expert

You use system variables in scripts and during configuration. System variables depend on the dialogs (Page 239), therefore you call the system variables directly in the dialog.

Most system variables are provided with values for the first time during configuration. These values can be modified when using configuration files for the deployment or executing the package. Some variables are only available during the package execution.

Also during the package execution, you can change system variables in dialog scripts or change orders by write access operations before these system variables become active in the sequence.

Conventions

Take the following conventions into account:

- Upper/lower-case is not relevant
- In compound names, the beginnings of words are written in uppercase for better legibility
- Values of enumerations are written in uppercase throughout

8.8.2 System variables for configuration

The following system variables are assigned to the configuration, and are addressed via the prefix "Up.$Pack".

For all of the system variables, the time until a change of the system variables makes sense and is still effective for the package execution is specified in the respective "Change effective until" column.

The system variables with the "readOnly" entry cannot be changed in the script.

Table 8-29 Configuration - Up.$Pack. (Dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Pack.				
DeployName	STRING	Read only	Package name is defined in the configuration.	Sample
ProdVersion	Version		The version of the CMC software is not configurable.	`6.3.0.0.1023`
DeployDir	STRING		Storage location (path) of the package: Specification of the path in which the package will be placed when deployed by Expert	`D:\Deploy`
DeployTarget	Enumeration DeployTarget (Page 383)		Runtime system of the package: Specification with which runtime system the package is deployed	`Up.$Pack.DeployTargets.WINDOWS`
ARC	BOOL		"Use archive" data area Note: Subsequent selection in the deployed package is not possible.	false / true
NCU			"Use NCU/PPU" data area Note: Subsequent selection in the deployed package is not possible.	
PCU			"Use PCU" data area Subsequent selection in the deployed package is not possible.	

8.8.3 Enumerations for configuration

Enumeration		Type	Change effective	Description
Enumeration, processing modes - DeployTarget Up.$Pack.DeployTargets.				
	LINUX	Enumerator value	readOnly	Runtime system Linux package (NCU)
	Windows			Runtime system Windows package (PC/IPC)

8.8.4 System variables for the dialogs

8.8.4.1 System variables for the dialogs

The following system variables are assigned to the dialogs, and are addressed via the prefix `Up.$Dialog`.

For all system variables, the time until a change of the system variables makes sense and is still effective for the package execution is specified in the "Change effective until" table column.

8.8.4.2 System variables for the Package configuration dialog

Table 8-30 "Package configuration" dialog - PackageConfig (Dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.PackageConfig.				
Activated	BOOL	readOnly	Activating the dialog processing, such as: • Configured in Expert • Specified by an "*.upcfg" file	false / true
ProcessMode	Enumeration ProcessMode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`
CfgFile	STRING		Configuration preselection	"`myMachine_01.upcfg`"

8.8.4.3 System variables for the Notes on the package dialog

Table 8-31 Dialog "Notes on the package" - PackageNotes (dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.PackageNotes.				
Activated	BOOL	-	Activating the dialog processing	false / true
ProcessMode	Enumeration ProcessMode (Page 390)		Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`

8.8.4.4 System variables for Access to target system dialog

Table 8-32 Dialog "Access to target system"- AccessData (dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.AccessData.				
ProcessMode	Enumeration ProcessMode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`
Target	Enumeration AccessData (Page 390)	Current dialog → OnNext	Enumeration selection of target system	`Up.$Dialog.AccessData.Targets.NCU`
NcuAddress	STRING		Address or IP address of the NCU/PPU	`"NCU1760"` `"10.8.8.126"`
NcuUserName			NCU/PPU login information: User name	`"manufact"`
IpcAddress			PCU address	`"192.168.214.241"`
IpcUserName			Access data PCU	"User"

8.8.4.5 System variables for the select offline archive dialog

Table 8-33 "Select offline archive" dialog - ArcSelection (Dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.ArcSelection.				
ProcessMode	Enumeration ProcessMode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`
ArchiveIn	STRING		Input archive archive file	`"D:\Temp\NC.dsf"`
NcFile			Path of the NC data	`"NC.dsf"`
PlcFile			Path of the PLC data	`"PLC-Daten.dsf"`
DrvFile			Path of the DRV data	`"DRV.dsf"`
HmiFile			Path of the archive file with HMI data	`"DSF_SinumerikOne_default_config_real.dsf"`
SysFile			Path of the archive file with system settings	`"DSF_SinumerikOne_default_config_real.dsf"`
ArchiveOut			Output archive archive file	`"D:\Temp\NC_Result.dsf"`

8.8.4.6 System variables for Backup initial state dialog

Table 8-34 Dialog "Backup initial state" - NcuOrigin (dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.NcuOrigin.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	false / true
ProcessMode	Enumeration ProcessMode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`
Backup	BOOL		Complete data backup (TGZ) before package run	false / true
Archive			NCPLCDP archive "Backup.dsf" before package run	

8.8.4.7 System variables for the PCU setup dialog

Table 8-35 "PCU setup" dialog - PCU setup (Dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.PcuSetup.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	`false / true`
ProcessMode	Enumeration ProcessMode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`

8.8.4.8 System variables for the CNC software dialog

Table 8-36 "CNC software" dialog - NcuSetup (Dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.NcuSetup.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	false / true
ProcessMode	Enumeration ProcessMode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`
Mode	Enumeration Mode (Page 390)		Enumeration installation mode	`Up.$Dialog.NcuSetup.Modes.INSTALL`
TgzFile	STRING		Preselection (CNC software (*.tgz))	`"./SW/std31a10c_hmi.tgz"`

8.8.4.9 System variables for the System configuration dialog

Table 8-37 "System configuration" dialog - SystemConfig (Dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.SystemConfig.				
ProcessMode	Enumeration ProcessMode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`
NcSource	Enumeration NcSource (Page 391)		Enumeration, origin of NC data	`Up.$Dialog.SystemConfig.NcSources.ORIGIN`
NcFile	STRING		Path of the NC data	`"DSF_SinumerikOne_default_config_real.dsf"`
PlcSource	Enumeration PlcSource (Page 391)		Enumeration, origin of PLC data	`Up.$Dialog.SystemConfig.PlcSources.ORIGIN`
PlcFile	STRING		Path of the PLC data	`"DSF_SinumerikOne_default_config_real.dsf"`
DrvSource	Enumeration DrvSource (Page 392)		Enumeration, origin of DRV data	`Up.$Dialog.SystemConfig.DrvSources.ORIGIN`
DrvFile	STRING		Path of the DRV data	`DRV.dsf`
HmiSource	Enumeration HmiSource (Page 392)		Enumeration, origin of HMI data	`Up.$Dialog.SystemConfig.HmiSources.ORIGIN`
HmiFile	STRING		Path of the archive file with HMI data	`"DSF_SinumerikOne_default_config_real.dsf"`
SysSource	Enumeration SysSource (Page 393)		Enumeration, origin of system settings	`Up.$Dialog.SystemConfig.SysSources.ORIGIN`
SysFile	STRING		Path of the archive file with system settings	`"DSF_SinumerikOne_default_config_real.dsf"`

8.8.4.10 System variables for the SINAMICS topology dialog

Table 8-38 "SINAMICS topology" dialog - DriveTopology (Dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.DriveTopology.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	false / true

System variables	Type	Change effective until	Description	Example
ProcessMode	Enumeration Process-Mode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`
AxisDriveAssignment	BOOL		Selection of the axis-drive assignment	false / true
UstFile	STRING		File name "*.ust" (comparison topology)	`"C:\Data\sample-comparison.ust"`
UtzFile			File name "*.utz2" (user-specified topology)	`"sample-userspecified.utz2"`

8.8.4.11 System variables for the Version Display dialog

Table 8-39 "Version Display" dialog - VersionView (Dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.VersionView.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	false / true
ProcessMode	Enumeration Process-Mode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`

8.8.4.12 System variables for the Step tree dialog

Table 8-40 "Step tree" dialog - StepSelection (Dialog ID)

System variables	Type		Description	Example
Up.$Dialog.StepSelection.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	`false / true`
ProcessMode	Enumeration Process-Mode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`
ArchiveBeg	BOOL		NCPLCDP archive `"StepBeg.dsf"` before step tree	`false / true`
ArchiveEnd		Current dialog → OnEnd	NCPLCDP archive `"StepEnd.dsf"` after step tree	

8.8.4.13 System variables for Backup result state dialog

Table 8-41 Dialog "Backup result state" - NcuResult (dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.NcuResult.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	`false / true`
ProcessMode	Enumeration ProcessMode (Page 390)	Current dialog → OnInit	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`
Backup	BOOL		Complete data backup (TGZ) after package run.	`false / true`
Archive			NCPLCDP archive "Result.dsf" after package execution.	

8.8.4.14 System variables for the Finish dialog

Table 8-42 "Finish" dialog - PackageEnd (Dialog ID)

System variables	Type	Change effective until	Description	Example
Up.$Dialog.PackageEnd.				
ProcessMode	Enumeration ProcessMode (Page 390)	Last script before dialog	Enumeration processing mode	`Up.$Dialog.ProcessModes.MANUAL`
LogDir	STRING		Path to storage location of the logbook.	`"D:\Log"`
LogName			Logbook file name.	`"CMC_Package_Log"`

8.8.5 Enumerations for dialogs

8.8.5.1 Enumerations

The enumerations are displayed in tables. The elements of an enumeration are listed in the table and are provided with brief descriptions.

8.8.5.2 Enumeration processing modes - ProcessMode

Enumeration		Type	Change effective	Description
Enumeration, processing modes - ProcessMode Up.$Dialog.ProcessModes.				
	AUTOMATIC	Enumerator value	readOnly	"Automatic" processing mode
	MANUAL			"Manual" processing mode
	PROGRESS			"Progress" processing mode

8.8.5.3 Enumeration selection of target system - AccessData

Enumeration		Type	Change effective	Description
Enumeration selection of target system - AccessData Up.$Dialog.AccessData.Targets.				
	NCU	Enumerator value	readOnly	Target system "NCU"
	IPC			Target system "NCU and PCU"
	VMC			Target system "Virtual machine"

8.8.5.4 Enumeration installation modes - Mode

Enumeration		Type	Change effective	Description
CNC software dialog - NcuSetup				
Installation modes enumeration - Mode Up.$Dialog.NcuSetup.Modes.				
	NONE	Enumerator value	readOnly	Installation mode without installation
	INSTALL			Installations mode "New installation"
	UPDATE			Installation mode "Upgrade"

8.8.5.5 Enumeration, origin of NC data - NcSource

Enumeration		Type	Change effective	Description
System configuration dialog - SystemConfig				
Enumeration, origin of NC data - NcSource Up.$Dialog.SystemConfig.NcSources.				
	ORIGIN	Enumerator value	readOnly	The actual data of the NC at the time of packet execution is used.
	FACTORY			The NC default data is used after a general reset.
	ARCHIVE			The data is taken from an archive.
	UNUSED			This indicates to the package that this data shall not be edited or processed.

8.8.5.6 Enumeration source of the PLC data - PlcSource

Enumeration		Type	Change effective	Description
"System configuration" dialog - SystemConfig				
Source of the PLC data enumeration - PlcSource Up.$Dialog.SystemConfig.PlcSources.				
	ORIGIN	Enumerator value	readOnly	The actual PLC data at the time of package execution is used.
	ARCHIVE			The data is taken from an archive.
	UNUSED			This enumeration value indicates to the package that this data is not processed.

8.8.5.7 Enumeration source of the DRV data - DrvSource

Enumeration		Type	Change effective	Description
"System configuration" dialog - SystemConfig				
Enumeration, origin of DRV data - DrvSource Up.$Dialog.SystemConfig.DrvSources.				
	ORIGIN	Enumerator value	readOnly	The actual drive data at the time of package execution is used.
	AUTOMATIC			Data after an automatic device configuration (as for SINUMERIK Operate) is used.
	TARGET			A user-specified topology is used.
	ARCHIVE			The data is taken from an archive.
	UNUSED			This enumeration value indicates to the package that this data is not edited or processed.

8.8.5.8 Enumeration, origin of HMI data - HmiSource

Enumeration		Type	Change effective	Description
"System configuration" dialog - SystemConfig				
Enumeration, origin of HMI data - HmiSource Up.$Dialog.SystemConfig.HmiSources.				
	ORIGIN	Enumerator value	readOnly	The actual data of the HMI archive portion is used at the time of package execution.
	ARCHIVE			The data is taken from an archive.
	UNUSED			This enumeration value indicates to the package that this data is not edited or processed.

8.8.5.9 Enumeration, origin of system settings - SysSource

Enumeration		Type	Change effective	Description
"System configuration" dialog - SystemConfig				
Enumeration, origin of system settings - SysSource Up.$Dialog.SystemConfig.SysSources.				
	ORIGIN	Enumerator value	readOnly	The actual data of the SYS archive portion is used at the time of package execution.
	ARCHIVE			The data is taken from an archive.
	UNUSED			This enumeration value indicates to the package that this data is not edited or processed.

8.8.6 System variables for the steps

8.8.6.1 Overview

The following system variables are assigned to the steps and are addressed via the prefix "`Up.$Step`".

For all system variables, the time until a change of the system variables is appropriate - and still effective for the package execution - is specified in the "Change effective until" table column.

Table 8-43 Steps

System variable	Type	Query as of	Change effective until	Description	Example
Up.$Step[?] (Page 394)					
Up.$Step[?].	OBJECT	**Dialog** Package configuration **Event** OnInit	readOnly	Step query whether the step with the specified ID exists. ""null"" is returned if the specified step ID does not exist. To determine the activation status, use the ".Activated" property.	`If Up.$Step[1337] != null`

System variable	Type	Query as of	Change effective until	Description	Example
Activated (Page 394)	BOOL	**Dialog** Package configuration **Event** OnInit	Last script before current step	The activation status of the specified step is returned. If the step is activated, then "true" is signaled. This is, the check box is set with a checkmark and thus activated.	`If Up.$Step[1337].Activated == true`
Locked (Page 396)				"Activation" locked status	`If Up.$Step[1337].Locked == true`
Collapsed (Page 397)				"true" is signaled if the substeps below the queried step have been closed. An operator cannot see the closed substeps.	`If Up.$Step[1337].Collapsed == true`

8.8.6.2 Query step

Query step - `Up.$Step[id]`

You can use the `Up.$Step[id]` system variable to query whether a specific step is available in scripts or dialog scripts.

Example

```
CHANDATA(1)
If Up.$Step[C_AXIS] != null          ; Step with ID: C_AXIS is available
  If                                 ; Step with ID: C_AXIS is activated
Up.$Step[C_AXIS].Activated == true
    $MA_IS_ROT_AX[AX5] = 1
  EndIf
EndIf
```

8.8.6.3 Activate step

Activate step - Up.$Step[id].Activated

The activation status of a specific step is defined in the step tree using the "`Up.$Step[id].Activated`" system variable.

The "`Up.$Step[id].Activated`" system variable supplies the value "true" for all steps whose check box are displayed with checkmarks.

The value "false" is supplied for all steps without a checkmark.

After a step is activated, the "Execute" property is automatically activated in the "Step configuration" property group.

Whether the step is actually executed or not depends on the upstream steps. The substep is not executed if a substep is activated, however the higher-level main step is not activated.

Example activate step

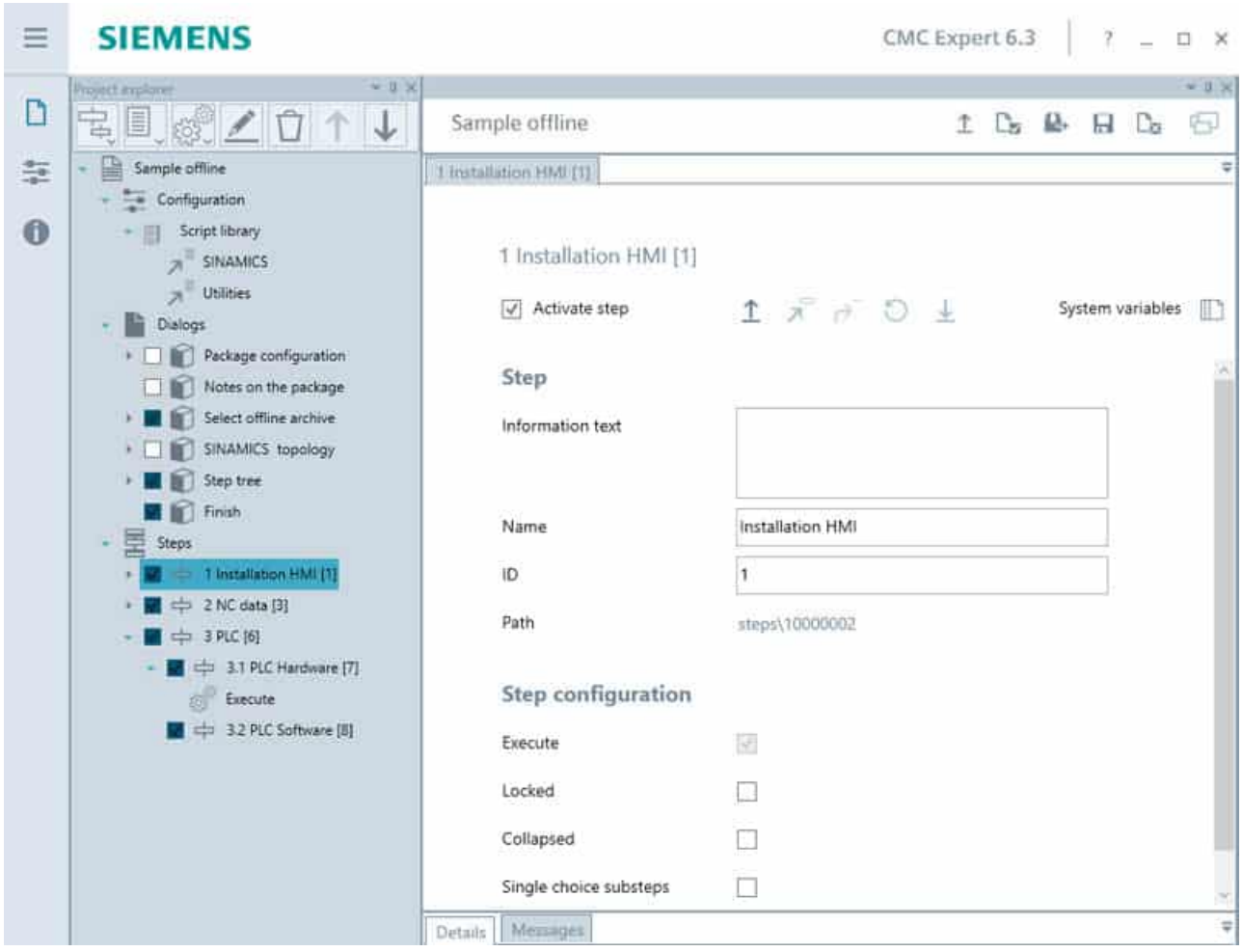


Figure 8-79 Example "Activate step"

You configure the "Step configuration" for each step individually via the check boxes, which you can activate or deactivate again.

Note

Step activation

You can configure step activation directly in the step tree. The step activation becomes immediately active.

8.8.6.4 Locking step activation

Step activation locked - Up.$Step[id].Locked

The activation status of a step is locked for package execution with the "`Up.$Step[id].Locked`" system variable.

The "Up.$Step[id].Locked" system variable supplies the value "true" for all steps whose check box are displayed with checkmarks.

Example of locked main step or higher-level step

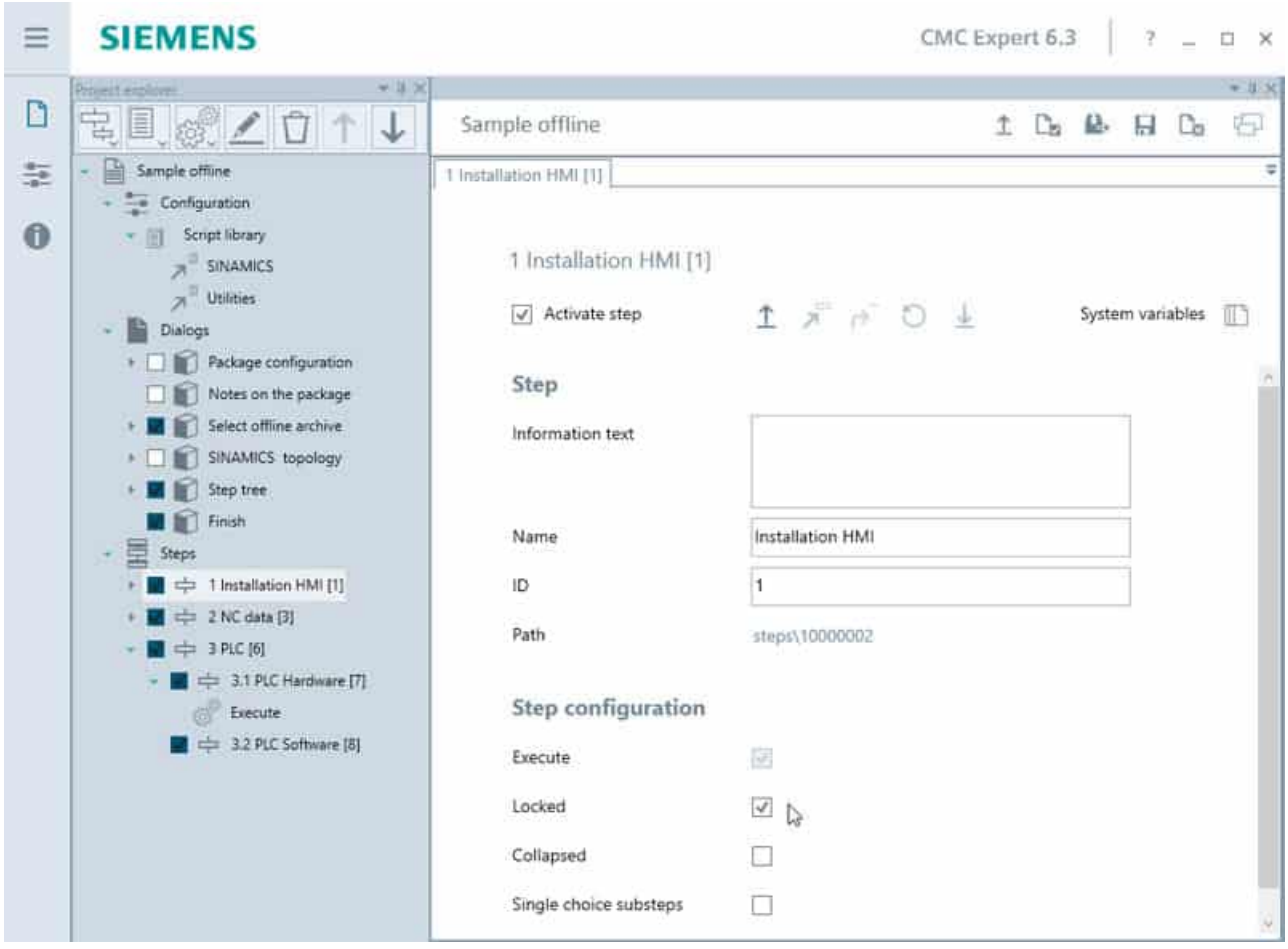


Figure 8-80 Example "Main step locked"

Note

Main step locked

If you remove the lock from the main step, the substeps will remain locked.

The locking of a substep must be undone for each individual step.

8.8.6.5 Collapse step

Collapsing steps - Up.$Step[id].Collapsed

The selected step is shown collapsed in the step tree with the "`Up.$Step[id].Collapsed`" system variable. The operator cannot see the step.

The system variable "`Up.$Step[id].Collapsed`" returns the value "true" for all steps whose check box for "Collapsed" is activated.

Example collapsing steps

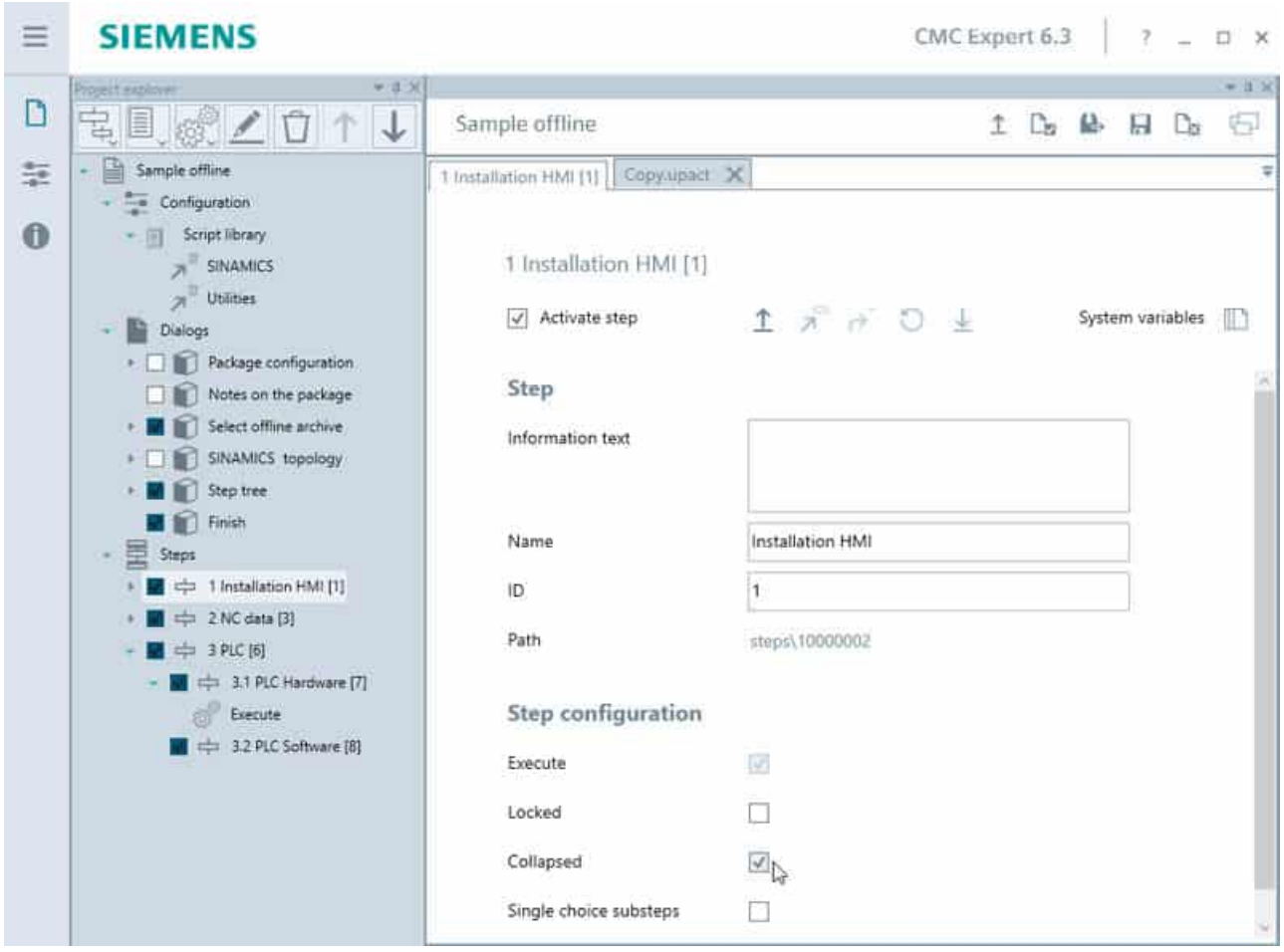


Figure 8-81 Example of "Step collapsed representation"

Note

Collapse steps

The "Collapsed" status only becomes effective when the package is executed. The user cannot have these steps displayed.

8.8.7 System variables for the environment

8.8.7.1 System variables for environment $Env

The subsequently listed system variables are used to query information from the environment of the running package execution. These system variables are accessed read-only (read only).

These system variables are accessed using prefix "Up.$Env".

System variables for the environment	Type	Description	Example
Up.$Env.			
RunTime	RunTimes	"RunTimes" enumeration	`Up.$Pack.DeployTargets.WINDOWS`
NCU	STRING	NCU/PPU type (STRING off "hwversion.xml")	"SIEMENS SINUMERIK ONE 1760"
PLC		PLC type (STRING off "hwversion.xml")	"SIEMENS SINUMERIK ONE 1760"
CFID		SD card ID (STRING off "hwversion.xml")	"S4GSG8JD41MQM6RN"
BatchMode	BOOL	The packages/script execution was started in the batch mode via the command line.	`false / true`

"hwversions.xml" on the SD card under "user/system/..."

8.8.7.2 Enumeration, runtime environments - Runtimes

Enumeration	Type	Description
Enumeration runtime environments - Runtimes Up.$Env.RunTimes.		
LINUX	Enumerator value	Executing the package under LINUX
WINDOWS		Executing the package under WINDOWS

8.9 Forwarding a project/creating a package

Deployment of a CMC package to a third party

Note

Legal information on Open Source Software and Commercial off-the-shelf

If a CMC package in which Shield is integrated with Open Source Software or Commercial off-the-shelf is deployed to a third-party, the CMC package must contain explicit information on the legal license conditions.

Security

NOTICE

Ensuring authorized access

By preconfiguring the access data the associated rights are transferred to the new owner or user of the respective files. These files include projects, packages, configuration files, change orders, etc.

The user must ensure with organizational means that only authorized persons have access to these files.

NOTICE

Administration and backup of project and package files

You are responsible for the security of the administration and the security of your project and package files.

Carefully protect your data, as manipulation in your project and package files can result in material damage.

More information on security can be found in the Industrial Security (https://support.industry.siemens.com/cs/ww/en/view/108862708) documentation:

- Section 7.1.9 Data backup
- Section 6.4 System Integrity

Note

Ensure security of the CMC packages

Handle CMC packages responsibly with regard to security aspects.

- Ensure that you handover valid CMC packages.
- Observe the security aspects when handling USB flash drives on which CMC packages are transported.

Procedure

Deploy the Expert projects that have been created as Windows or Linux packages. These packages are then executed under the respective operating system.

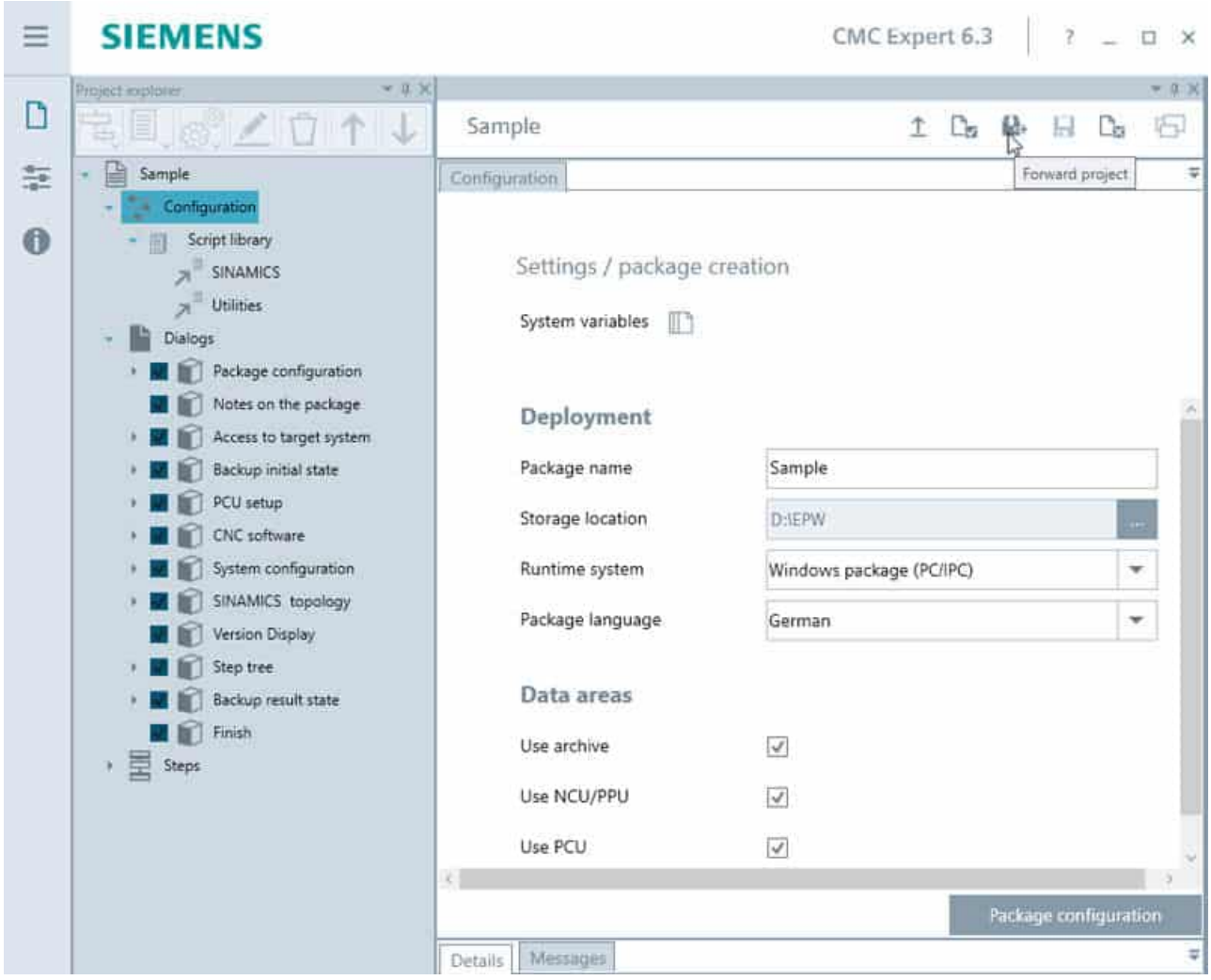


Figure 8-82 Deploy project

To create a package, proceed as follows:

1. Click on "Deploy project". You have already defined whether a "Windows package (PC/IPC)" or a "Linux package (NCU)" is deployed in "Project explorer" in the "Configuration" under "Deployment" > "Runtime system".
2. In the newly opened "Deploy project" dialog, you define the "Deployment folder" and the "Deployment name" of the project.
3. Click on "Continue".
   An automatic check run, which checks the project for inconsistencies and errors, is started.
   You can also manually execute the check run using button "Review project".
   A package is only created if the project check run did not identify an error.

Note

Note for the Linux package

In a Linux package, no file or path information with umlauts in the file names or folder names can be used on the NCU.

8.10 Package execution

8.10.1 Requirements

Create MyConfig - executing packages

Note

Responsibly handling packages

For Windows and Linux systems, you can deploy Create MyConfig packages using Expert.

For Linux operating systems running on the NCU/PPU, Expert generates a file with the extension "*.usz" (Linux package).

For Windows operating systems running on the IPC, PG or PC, a file with the extension "*.exe" (Windows package) is generated.

The following sections describes the procedure for the Windows and Linux operating systems.

Package execution using NCU/PPU

Observe the following points if an NCU/PPU is involved during package execution (NCU/PPU data area active):

- To prevent errors during package execution, the NCU/PPU system state must be consistent.
- Before starting a CMC package, the NCU/PPU must be in the following state:
  - LEDs "RDY" and "RUN" are lit green
  - The 7-segment display indicates a "6" with flashing point
- If system software is on the SD card, then the following must apply before starting a package:
  - The NC status display is at "6"
  - The PLC status display is "green".

  Use a service stick for an SD card without system software.
- Load the factory settings prior to a drive unit configuration if inconsistent data is found on the control. Failure to adhere to this can lead to undefined conditions.
- For a drive unit configuration - automatic configuration or user-specified topology - carefully ensure that the PLC hardware configuration is consistent with the hardware used regarding the NCU and NX configuration.
  The PLC hardware configuration may contain more NX modules than are available on the machine.
- The SINAMICS data must be consistent and match the hardware configuration.
- If you renumber the drive objects, then you are responsible for adapting the BICO interconnections.
  The BICO interconnections are not adapted automatically by CMC.
- On the "Finish" dialog page, you must ensure that the package has been executed without any errors.
  If processing took place with errors, you must check the log file as to whether there will be any likely effects on functionality.

If this state is not present before the start of a CMC commissioning package, then it can be achieved by a memory reset and loading of default data of the individual control components.

Procedure, memory reset / load standard data

For NCU/PPU, proceed as follows:

1. NC: S3 = 1 and OFF/ON.
2. Restart the device.
3. When running up, wait for the "6" with flashing point. It can take several minutes for the system to run-up.
4. Then "S3" back to "0".
5. SINAMICS: Empty the folder "user/sinamics/data" on the SD card followed by ON/OFF.

8.10.2 Executing the package under Windows

8.10.2.1 Executing a standard package

Starting to execute the package under Windows

Establish a network connection between the PC and the control devices (NCU17xx, IPC) that are to be processed with the package.

You can start the package created by Expert for Windows with the file extension "*.exe" on the IPC, the PG or the PC with "Execute" - or with a double-click.

If the package file is the only exe file in the "Install" folder of a USB medium inserted in the IPC (USB FlashDrive, hard disk, CD), then the file is automatically started when the IPC is run-up.

Note

Simultaneously starting several packages

Carefully note that it is not permissible to simultaneously start several packages.

A package may only be started if no other package is running.

Note

Package names

Avoid the designations "install", "installation" or "setup" in the names of packages.

These designations result in confirmation prompts from Windows and unwanted error messages on the IPC.

Note

Unknown USB medium

For existing IPC base systems, ensure that the IPC recognizes the USB medium, as a new medium is installed after the start routine for the package. This means that it may be necessary to insert the USB medium and then switch-off and switch-on the IPC again after approximately 30 seconds.

8.10.2.2 Package execution dialogs

The package execution dialogs can be seen in the section under "Dialogs (Page 239)".

Package execution is shown in the descriptions for the dialog pages in the "Dialog view while executing the package (for commissioning engineers / operators)" section.

8.11 Command line call

8.11.1 Package deployment by Expert

To deploy a package, you can start Create MyConfig Expert in the Command line mode. The command line call is intended for the automated deployment of packages.

For deployment in the command line mode, specify command line parameters (switches); these parameters may be optional.

Call

Write the command subsequently shown as well as all associated switches and arguments in one line to a batch file or a command line interface.

Deployment with check run

```
ExpertCmd.exe deploy [-d <Deployment folder path>] [-n <Package
name>] [-t <Target system>] [-a <Batch mode>] [-x <Log file path>]
<Project file path>
```

The switches in square brackets are optional.

Example:

```
ExpertCmd.exe deploy -d "D:\packages" -n "package" -t linux -a
AlwaysYes -x "D:\Log\L1 V1.xml" "D:\Projects\MyProject
\MyProject.upproj"
```

Notes

Note the following information for the command line call:

Note

Specification of the complete path of "ExpertCmd.exe"

Specify the complete path of the "ExpertCmd.exe", or include the path in the environment variable "Path".

Example:

"C:\Program Files\Siemens\Automation\SINUMERIK\Create MyConfig 6.3\ExpertCmd.exe"

Note

Path data

- Set the specified paths in straight quotation marks if spaces are included.
  Example:
  "C:\path with spaces\Expert_Logfile.xml"
- When specifying folders, path data may only end with `'\'`, e.g. `-d "D:\packages\"`

Note

Saving batch files

Always save batch files with ANSI/ASCII coding.

Note

Umlauts and special characters

Avoid the use of umlauts and special characters in path names.

To allow special characters or umlauts to be correctly processed in specified paths, save the batch files with command line calls with the same character set as used by the region-specific command line interpreter "cmd.exe".

For example, "Codepage 850", also known as "DOS-Latin-1", applies to the region Western Europe.

The "Notepad" editor integrated in Windows does not support saving files with this character set.

Note

Uppercase/lowercase

For the command line switches and arguments, **always** use the specified uppercase/lowercase syntax.

Note

Absolute/relative path

You can specify the absolute path or the relative path.

The relative path with `".\"` or `"./"` refers to the current work folder, where the batch or CMD was called.

If no path is specified, the work folder is used.

Parameter

Significance of the switches

Package deployment	
`Usage: deploy [options] <projectfile>`	
Arguments:	
`projectfile Expert project (.upproj)`	
Options:	
`-h|--help`	Display Help

Package deployment	
`-d|--deploydir`	Deployment folder Default: as specified in the project.
`-n|--deployname`	Package name Default: as specified in the project
`-t|--target`	Target system: Windows or Linux Default: as specified in the project.
`-a|--answer`	Automatic batch mode Response to each question with the specified value Default: Interactive
`-x|--xmllog`	Write XML report

Return values

Note

Please observe "ExpertCmd"

The return values for the command line call are only applicable for "ExpertCmd.exe".

The return values from ExpertCmd are determined by querying the "%ERRORLEVEL%" variables.

The following example shows how the "%ERRORLEVEL%" variables are used.

Table 8-44 Return values Expert

Value	Description
0	No errors or alarms occurred when deploying the package.
1	Alarms but no errors occurred when deploying the package.
2	Errors (and possibly alarms) occurred when deploying the package.
3	Incorrect parameter specified
31	Invalid target system
32	No project file available
33	Invalid project

Value	Description
4	Runtime error
41	Target directory invalid

Example

Assumption: Create MyConfig is installed in the "C:\Program Files\SIEMENS\Automation \SINUMERIK\Create MyConfig 6.3" directory and a "Pack_1.upz" package is available in the "C:\Packages\" path.

```
@echo off
PATH=%PATH%;"C:\Program Files\SIEMENS\Automation\SINUMERIK\Create MyConfig
6.3"

ExpertCmd.exe deploy -d "D:\packages" -n "package" -t linux -a AlwaysYes -x
"C:\LogM1.xml" "C:\Packages\Pack1\Pack1.upproj"

if %ERRORLEVEL% EQU 3 (
    @echo "Error when calling ExpertCmd.exe!"
    pause
)
if %ERRORLEVEL% EQU 2 (
    @echo "Error during transmission!"
    pause
)
if %ERRORLEVEL% EQU 1 (
    @echo "Alarms during deployment!"
    pause
)
if %ERRORLEVEL% EQU 0 (
    @echo "Deployment executed without error!"
    pause
)
```

See also

Check run (Page 407)

8.11.2 Check run

For the check run of a package, you can also start Create MyConfig Expert in the command line mode. The command line mode is intended for the automated check run of packages.

Specify command line parameters (switches) for the check run in command line mode. The command line parameters are partly optional.

Call

Write the command subsequently shown as well as all associated switches and arguments in one line to a batch file or a command line interface.

Check run when Expert opens

```
ExpertCmd.exe verify [-x <Log file path>] <Project file path>
```

The switches in square brackets are optional.

Example:

```
ExpertCmd.exe verify -x "D:\Logs\MyProject.xml" "D:\Projects
\MyProject\MyProject.upproj"
```

Notes

Note the following information for the command line call:

Note

Path data

- Set the specified paths in straight quotation marks if spaces are included.
  Example:
  "C:\path with spaces\Expert_Logfile.xml"
- Paths must never end with '\'.

Note

Saving batch files

Always save batch files with ANSI/ASCII coding.

Note

Umlauts and special characters

Avoid the use of umlauts and special characters in path names.

To allow special characters or umlauts to be correctly processed in specified paths, save the batch files with command line calls with the same character set as used by the region-specific command line interpreter "cmd.exe".

For example, "Codepage 850", also known as "DOS-Latin-1", applies to the region Western Europe.

The "Notepad" editor integrated in Windows does not support saving files with this character set.

Note

Uppercase/lowercase

For the command line switches and arguments, **always** use the specified uppercase/lowercase syntax.

Note

Absolute/relative path

You can specify the absolute path or the relative path.

The relative path with ".\" or "./" refers to the current work folder, where the batch or CMD was called.

If no path is specified, the work folder is used.

Parameter

Significance of the switches

Check run	
`Usage: verify [options] <projectfile>`	
Arguments:	
`projectfile Expert project (.upproj)`	
Options:	
`-h|--help`	Display Help
`-x|--xmllog`	Write XML report

Return values

Note

The return values for the Expert command line call apply only if "ExpertCmd" is executed.

The return values from "ExpertCmd" can be determined by querying the "%ERRORLEVEL%" variables.

The following example shows how the "%ERRORLEVEL%" variables are used.

Table 8-45 Return values "ExpertCmd"

Value	Description
0	No errors or warnings occurred when validating the package.
1	Warnings but no errors occurred when validating the package.
2	Errors (and possibly warnings) occurred when validating the package.
3	Incorrect parameter specified
31	Invalid target system
32	No project file available
33	Invalid project
4	Runtime error
41	Target directory invalid

Example

Assumption: Create MyConfig is installed in the "C:\Program Files\SIEMENS\Automation \SINUMERIK\Create MyConfig 6.3" directory and a "Pack_1.upz" package is available in the "C:\Packages\" path.

```
@echo off
PATH=%PATH%;"C:\Program Files\SIEMENS\Automation\SINUMERIK\Create MyConfig
6.3"

ExpertCmd.exe verify "C:\Packages\Pack_1\Pack_1.upproj"

if %ERRORLEVEL% EQU 3 (
    @echo "Errors when calling Expert.exe!"
    pause
)
if %ERRORLEVEL% EQU 2 (
    @echo "The check run reported errors!"
    pause
)
if %ERRORLEVEL% EQU 1 (
    @echo "The check run reported warnings!"
    pause
)
if %ERRORLEVEL% EQU 0 (
!"The check run reported no errors and no warnings    @echo "
    pause
)
```

8.11.3 Package execution

Call

Write the command subsequently shown as well as all associated switches and arguments in one line to a batch file or a command line interface.

The significance of the switches and arguments is described in the following Table "Package execution parameters".

```
<Package> [-hide] [-batch]
```

Notes

Note the following information for the command line call:

Note

Observe the sequence of the switches

Write the switches in the specified sequence.

When doing this, observe the "Call" section and the "Significance of the switches" table.

Note

Saving batch files

Always save batch files with ANSI/ASCII coding.

Note

Umlauts and special characters

Avoid the use of umlauts and special characters in path names.

To allow special characters or umlauts to be correctly processed in specified paths, save the batch files with command line calls with the same character set as used by the region-specific command line interpreter "cmd.exe".

For example, "Codepage 850", also known as "DOS-Latin-1", applies to the region Western Europe.

The Notepad editor integrated in Windows does not support saving files with this character set.

Note

Uppercase/lowercase

For the command line switches and arguments, **always** use the specified uppercase/lowercase syntax.

Note

Absolute/relative path

You can specify the absolute path or the relative path.

Parameter

Table 8-46 Package execution parameters

Switch	Arguments	Description
-hide	-	• This causes the package to be processed without being displayed. • Implies option "-batch".
-batch	-	• This causes the package selection and the completion message to be bypassed for packages with NCU configuration. System variable `Up.$Env.BatchMode` is set.

Return values

Table 8-47 Return values of the package execution

Value	Description
0	No error has occurred
1	Canceled by the user
2	Canceled by step script or script function
3	Incorrect command line call.

Example

```
paket.exe -batch
```

Create MyConfig - packages

9

9.1 Starting the package execution

Information about the package execution is provided in Expert in Section "Package execution (Page 401)".

9.2 Package execution dialog pages

The dialog pages of the package execution are described in Expert in Section "Dialogs (Page 239)".

9.3 Command line call

Information on the command line call is provided in the section with the same name under Expert (Page 404).

Create MyConfig - Topo

10

10.1 Introduction

Note

Documentation structure: Operating Manual / Online Help

Section "Topo" has the same layout and structure in the Operating Manual and online help.

What is Topo?

The "Topo" software is used in Create MyConfig to configure, display, print and compare SINAMICS topologies.

The following topology variants are available.

- Comparison topology
- User-specified topology

Comparison topology

An image of the SINAMICS topology is used for purposes such as:

- Check the topology (wiring) of SINAMICS on the machine
- Monitor module/component properties
- Change module/component names
- Change DO numbers and DO names
- Assign SINAMICS modules DO variables.

SINAMICS modules or components are, for example, Sensor and Motor Modules.

User-specified topology

Instead of using "Automatic configuration", for SINAMICS, specify an offline (target) topology package (SOT). This device commissioning variant supports a step-by-step production workflow, because modules/components and data that do not physically exist can be configured in the control.

Topo functions

Topo provides the following functions:

- Creation of topologies via a graphical user interface using the shortcut menu and additional standard Windows tools.
- Import of topologies from SINUMERIK archives, from deployed topologies (ECAD XML files)
- Import of topologies from Siemens Data Exchange "*.sdx" from EPLAN Add-In

- Check run for created topologies
- Comparison of topologies
- Assignment of module/component properties, e.g. component name, OEM ID.
- Specification of criteria (Article No.) that the package compares with the actual properties of the SINAMICS modules on the machine to be produced (read-out TARGET topology of the SINAMICS). Impermissible deviations are reported.
- Assignment of the DO variables to modules/components
- Creation of a user-specified topology based on generic modules.
  You can load the user-specified topology directly to SINAMICS when the package is being executed.
- Support of a step-by-step commissioning for missing components through deactivation of these components.
- A "DOList Editor (Page 436)" is integrated in Topo, which provides effective support when creating DO variables.
  If you want to edit DO variables "*.uvdx" independent or outside of Topo, then start the "DOList-Editor" via the Windows start menu as a single application.

Assignment of DO variables and DO properties

You assign DO variables to the modules in Topo. The DO variables provide properties that can be used for configuring the package execution in Expert. You can therefore work flexibly in Expert with defined DO variables, whose absolute values only apply during the package runtime on the machine.

You assign SINAMICS modules that do not generate their own DO, to DO-generating modules using DO variables, e.g. "Sensor Module" to "Motor Module".

You can therefore save drive-specific parameters in a modular fashion in a package without reference to a particular "DO number", "module number" or "slave".

The assignment of the parameters to the topology-specific DO on the machine is controlled flexibly in the scripts using DO variables corresponding to the package configuration.

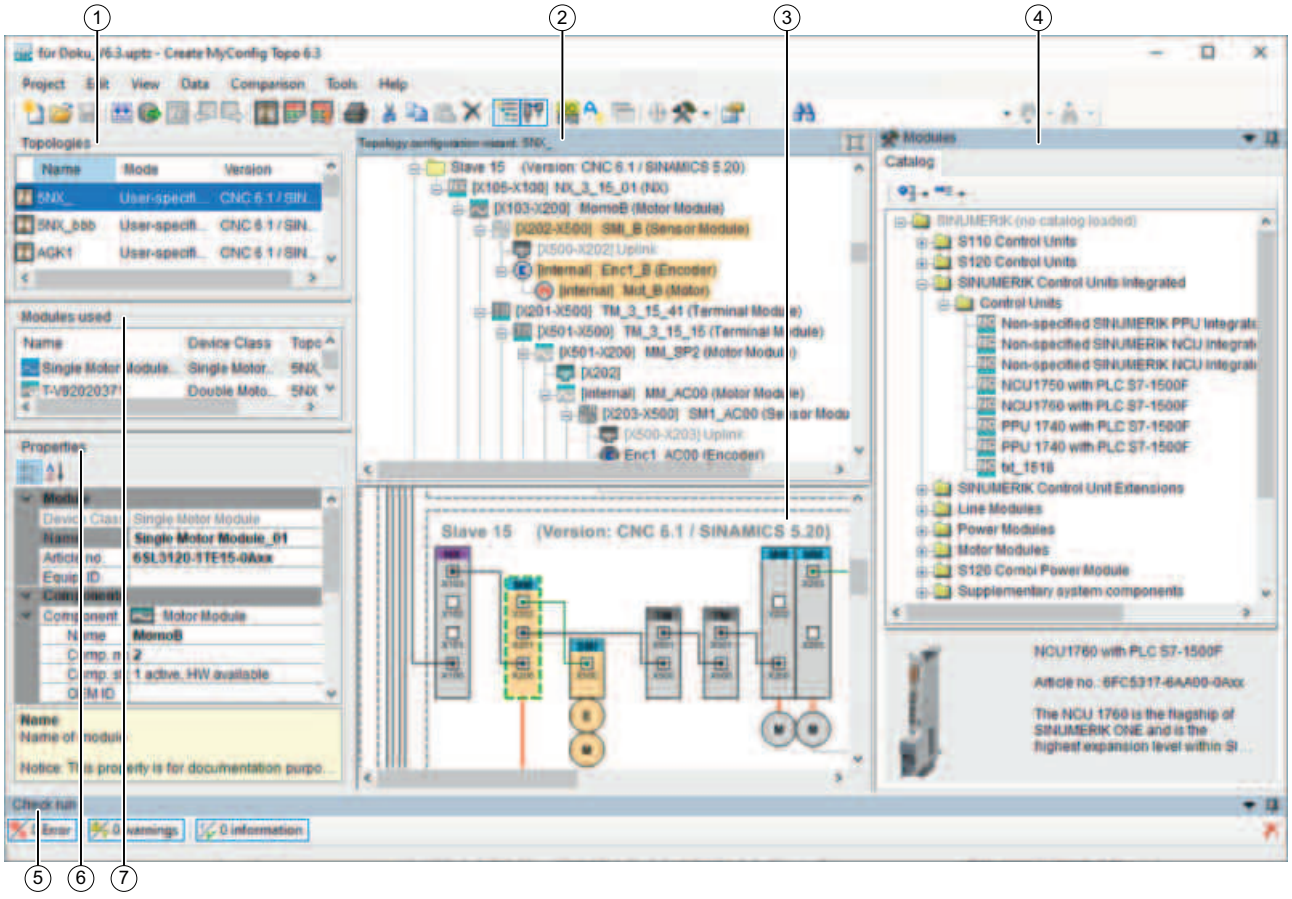
The DRIVE-CLiQ wiring created in Topo and the assignments made for DO variables, DO names and DO numbers are saved using a deploy function to a "*.ust" comparison topology or "*.utz2" user-specified topology file.

When executing a package, the existing file of the comparison topology or user-specified topology is accessed.

10.2 Overview of user interface

10.2.1 Topo - Overview window

The Topo software interface provides the following operating areas:



① Topologies
② Topology tree
③ Topology graph
④ Modules
⑤ Check run
⑥ Properties
⑦ Modules used

Figure 10-1 Overview of Topo user interface

10.2.2 Topo - menu functions

Topo provides the following main menus and submenus:

Table 10-1 Topo - Project main menu

Project			
	New	<Ctrl+N>	Creates a new project
	Open...	<Ctrl+O>	Opens an existing "*.uptz" project and all specified file types that can be imported.
	Save	<Ctrl+S>	Saves the current project as "*.uptz" file. If components are changed, they are marked in the topology tree with an asterisk at the component. The diskette icon is active to show that changes still need to be saved.
	Save as...		Saves the current project with a different path or file name.
	Check run	<Ctrl+B>	Checks an open topology regarding the correct allocation of DO variables and component numbers. **Note** Note that no verification is made as to whether the wiring is permissible according to SINAMICS guidelines.
	Deployment	Display the opened topologies with specification of the associated topology: • User-specified topology • Comparison topology	Deployment of an open topology in an "*.ust" (comparison topology) or "*.utz2" (user-specified topology) file that is used by the package. Before deployment, Topo first performs a topology check.
	Load topology to the control	Display of opened topologies	The finished project can be loaded to the control via a network connection. A configured package is executed in the background. Note that this is not an archive. The topology can be loaded to the control **only** with the "Extended user-specified topology". Make sure that source data, e.g. archives or ASCI data sets, is not generated from a software version that is newer than the target software version.

Project		
Display/import topology		**Displaying the topology**:
	From topology...	The following topologies are displayed: • Topology created by an electrical design system according to the ECAD scheme "*.xml" • Exported "*.ust" • Topology extracted from a drive archive • Topology taken directly from a control • Topology created using the EPLAN electric P8 Add-In - Siemens Data Exchange "*.sdx" **Import topology**: An external topology is opened write-protected and is not imported into the actual project. Clicking the "Display/import topology" button in the topology configurator imports the external topology into the current project. **EPLAN / Siemens Data Exchange**: The import from a Siemens Data Exchange "*.sdx" file is only available when you have installed the "ECAD Import Lizenz" in the Automation License Manager. If the SDX file does not contain a DO list, when importing from the Siemens Data Exchange "*.sdx" format via a dialog, a prompt is made to specify an appropriate "*.uvdx" file. Otherwise only the component names and DO names contained in the SDX file are imported and DO variables generated. When importing a topology from an SDX file, it can also be selected whether the configured component names should be augmented with the equipment identifier specified for modules in EPLAN. This simplifies diagnostics on the controller. The resulting component name corresponds to the following scheme: <Equip. ID>; <Component name>
	From archive...	You can import the topology from the following archive files: • (*.dsf) • (*.arc) • (*.ard)
	From control	Establish a network connection from the PC by specifying the IP address. An archive from the controller is then created and transferred to the PC. The archive is stored on the PC and opened there.

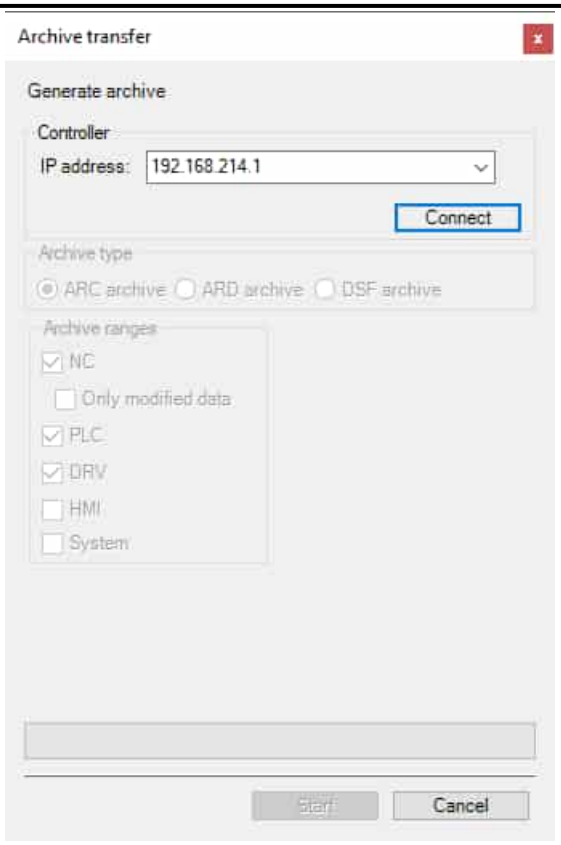
Project			
			
	Print topology		Prints either the topology tree or the topology graph.
	Properties...		Shows important project properties, such as path of the project, date of change, changed by, version and comment.
	Last opened		Shows the most recently opened projects; they can be directly selected.
	Displayed last		Selects a project from the last displayed projects.
	Close all		Closes the open project
	Exit	<Alt+F4>	Exits the "Create MyConfig Topo" program.

Table 10-2 Topo - Edit main menu

Edit			
	Cut	<Ctrl+X>	Removes the selected components and copies them to the clipboard.
	Copy	<Ctrl+C>	Copies the selected components to the clipboard.
	Paste	<Ctrl+V>	Pastes the components from the clipboard to the selected position.
	Delete	<Del>	Deletes the selected components and any following components connected to the component.

Table 10-3 Topo - View main menu

View			
	Topology tree		Shows or hides the topology tree. The topology tree and the graphic representation can never be hidden simultaneously.
	Topology graph		Shows or hides the graphic topology view. The topology tree and the graphic representation can never be hidden simultaneously.
	Zoom	25 % ... 300 %	Zooms the topology graph to a desired working size.
	Reset graphic		Resets the topology graph to the last saved working size.
	Display modules used		Shows the group titles for the modules used for the various device classes. You can show or hide the display.
	Groups in modules used		Shows in the "Modules used" window the group titles for the various device classes. You can show or hide the display.
	Toolbars	Standard	Shows or hides toolbars.
		Quick search	

View			
	Go forward to next error/ warning	<F8>	Errors or warnings can be displayed one after another.
	Go back to previous error/ warning	<Shift+F8>	Previous errors or warnings can be displayed.

Table 10-4 Topo - Data main menu

Data			
	Manage DO list...	Ctrl+Alt+D	Opens the "DO list" dialog for the administration of DO variables. The DO variables can be added, imported, exported, or linked. More information is provided in the section under "Manage DO list (Page 439)".
	Manage axis-drive assign-ment...	Ctrl+Alt+A	Opens the "Axis-drive assignment - Topology" dialog in which the axis-drive assignment table is shown and an axis-drive assign-ment can be performed. A special feature is the display of the automatically generated DO variables, which can also be assigned manually. When moving drive objects (DO), the required calculations are performed automatically. You can also perform the desired Axis-drive assignment manual-ly. More information is provided in the section under "Axis-drive assignment".

Table 10-5 Topo - Comparison main menu

Comparison			
	Refresh	Ctrl+D	If changes are present, the icon is shown in color and indicates that an update is required.
	Edit configuration...	Ctrl+M	Opens the "Configuration - Topology comparison" dialog in which you can manage comparison configurations (create, edit, delete, etc.).
	Configuration selection	Topology Complete	Compares the currently displayed topologies and switches over into the compare mode. More information is provided in the section under "Topology comparison (Page 451)". To display several topologies, select these in the list of topologies using the <Ctrl> key and a mouse click. A maximum of 2 topologies can be compared simultaneously.

Table 10-6 Topo - Tools main menu

Tools		
	Options...	Opens the "Options" dialog. You can make basic settings here, such as language and color changes to the graphical representation, in the following tabs: • Topology graph • Axis-drive assignment • Messages • Language

Table 10-7 Topo - Help main menu

Help			
	Content...	<F1>	Display of the online help for the **Topo** program. The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	FAQs...		Display of the Online Help "FAQs" The online help is identical to the section of the same name in the SINUMERIK Create MyConfig - Diff, Expert, Topo Operating Manual.
	About...		Contains information on the version and copyright for Create MyConfig Topo.

10.3 Topo - operation

10.3.1 Creating a topology

Topo offers the option of creating a new topology from the "module catalog". However, you can import and edit a topology from an electrical design system or directly from the control via an archive.

The topology can also be displayed from a drive archive, imported, or directly imported from a control and subsequently edited. In this case, the original component numbers and article numbers determined by SINAMICS are entered in the properties of the components. New DO variables are also automatically generated and assigned to the imported components.

The DO variables used for CNC commissioning are stored in the control and can be assigned and displayed again.

Representation types Topology tree / Topology graph

A topology is represented in 2 ways in Topo.

- In a hierarchically structured tree representation
- In a systematic, graphic representation

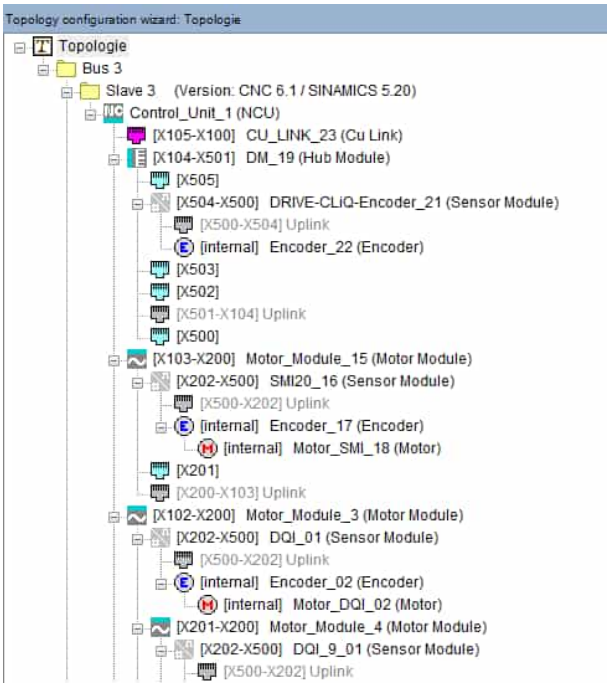
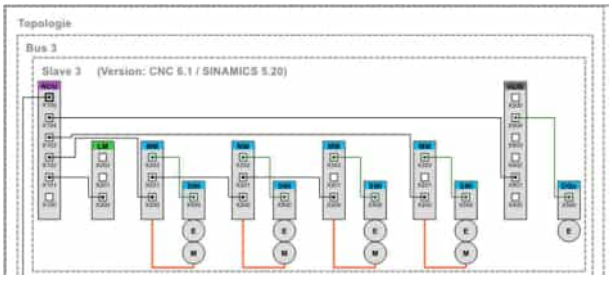
Both representation types make it easy to understand the wiring and identify the individual components.

As a consequence, both representation types offer the same functionality with regard to creating and editing a topology. As a consequence, you can work with one or both of these display forms.

The topology tree or the topology graph are displayed or hidden using menu "View" >"Topology tree" and "View" > "Topology graph".

To display and edit several topologies, you can select these from the list of topologies by making a multiple selection. The same as for "Topology comparison (Page 451)", both topologies are displayed next to one another.

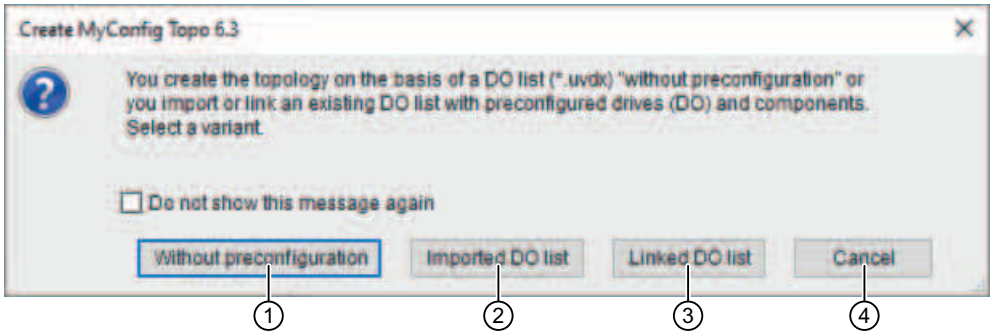
Table 10-8 Displaying the topology in the topology configurator: Topology tree and topology graph

Topology tree	Topology graph
 Example of a topology tree	 Example of a topology graph

Procedure for creating a topology

1. Start Topo.
2. To create a new project, select "Project" > "New" in the main menu.

3. The following window for creating a topology is displayed:



① **Without preconfiguration**

An empty topology opens.

② **Imported DO list**

An empty topology is opened, in which you can import a DO list "*.uvdx" from any drive.

③ **Linked DO list**

An empty topology is opened, in which you can link a DO list "*.uvdx" from any drive.

④ **Cancel**

The operation is canceled. Project and topology are not created.

Figure 10-2 Topology selection

After selecting a topology version, a new topology is automatically created with default name "Topology" and a "Bus 3" inserted.
If one of the versions based on a DO list is selected, the DO list contains preconfigured DO properties and component properties. This simplifies the assignment of properties when configuring the topology.

4. You can now modify the name of the topology and wire the "Bus 3" directly to an NCU. The slave is created automatically.

Note

Wiring of an external CU, e.g. CU320

Note that these cannot be connected to the internal "Bus 3".

First insert the appropriate bus 1, 2, 4 ... in the topology from the shortcut menu.

5. The components are wired either from the selected structure node of the tree view or directly from the wiring diagram via the "Connect component" shortcut menu entry.
   You can select the components that can be used at this point of the topology from the module catalog, and insert and wire in the list of the associated components. This ensures that a valid topology is created and that components are not connected twice.
   Alternatively, you can drag the components from the module catalog, drop them into the "Topology configuration wizard" and connect them.

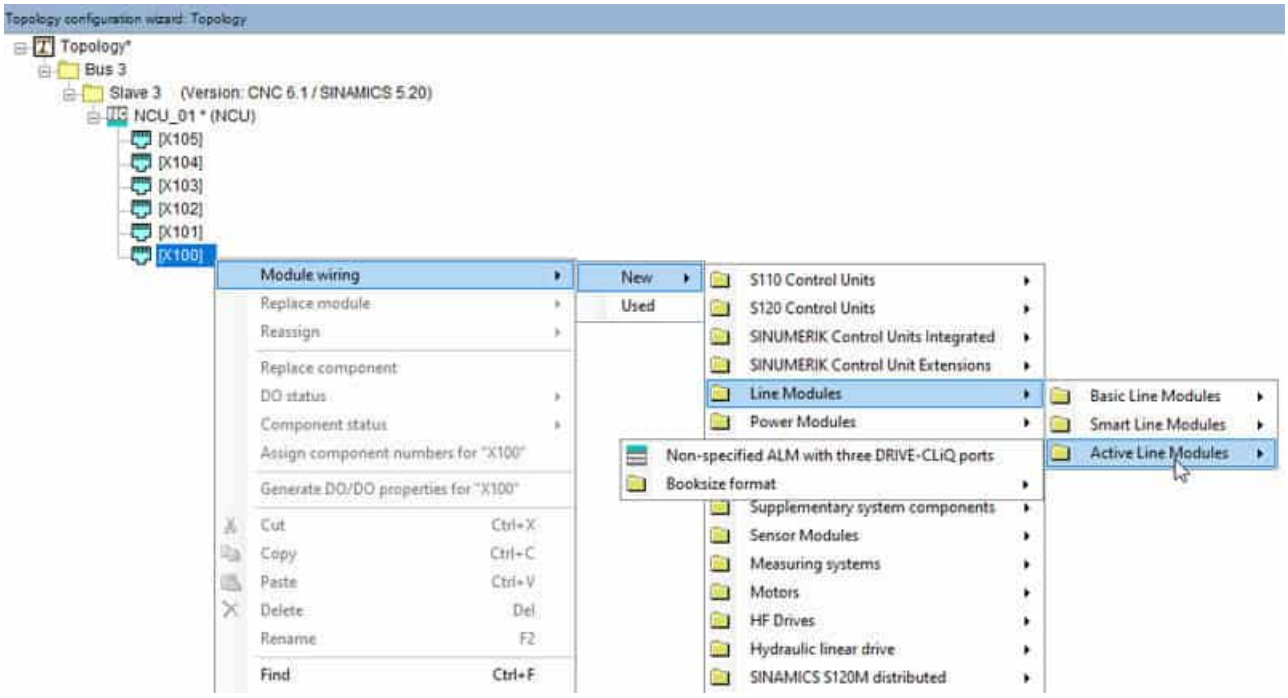


Figure 10-3 Creating a topology

6. The DO variables are assigned to the components in the component properties under "DO". Before you can make this assignment, you must have created or imported the DO variables. More information is provided in Section "Manage DO list (Page 439)".

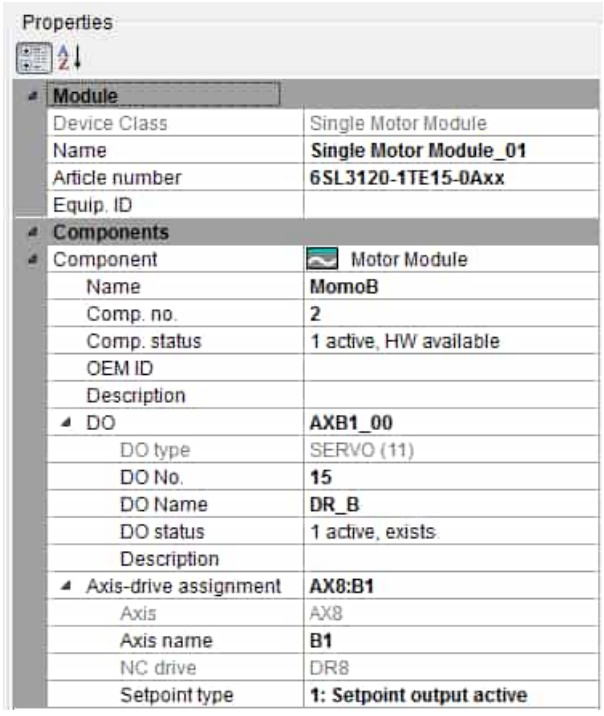


Figure 10-4 Topo properties

Additional functions

In addition to the standard functions such as "Cut", "Copy", "Paste", etc., the shortcut menu of the topology configurator provides further functions:

Table 10-9 Shortcut menu in the "Topology configurator": Topology tree / topology graph

Function			Description
	Wire module	New	Wires a new or already used module.
		Used	
	Replace module		Replaces an already wired module with an equivalent module from a different topology of the opened topology project. Only modules from the same device class that have the same number of DRIVE-CLiQ ports are listed. When replacing modules, all original wirings to further modules are retained.
	Convert module	NCU 4 DQ ports	Convert the existing modules into the required module type.
		NCU 6 DQ ports	
	Rewire	Downlink port	As standard, a module in Topo is always wired with its 1st DRIVE-CLiQ-Port (X100, X200, ...) to the lower-level module. If the connection is to be established to another port, then here, select any free DRIVE-CLiQ-Port.
		Uplink port	As standard, a module in Topo is always wired with its 1st DRIVE-CLiQ-Port (X100, X200, ...) to the higher-level module. If the connection is to be established to another port, then select any free DRIVE-CLiQ-Port. Rewiring is also possible by dragging & dropping.

Function			Description
	Replace component		Replaces the selected component with a preconfigured and unused component for the assigned DO in the DO list. The wiring of the component remains unchanged. The properties that do not belong to the preconfigured component also remain unchanged.
	DO status	0 = inactive, available 1 = active, available 2 = not available	Changes the "DO status" of the drive object of the selected component.
	Component status	0 = inactive, HW available 1 = active, HW available 2 = HW not available	Here, change the "Component status" of the selected component.
	Assign component number		Assigns the component numbers for the selected component or for all lower-level components of the selected object. Select whether only missing component numbers are augmented or all component numbers are automatically regenerated.
	Generate DO/DO attributes		Generates DO or supplements missing DO properties for the selected component or for all lower-level DO-forming components of the selected object. In the dialog, you select between the options: • Add missing DOs and properties • Generate new DOs and properties
	Cut	<Ctrl+X>	Cuts the selected object and copies it to the clipboard.
	Copy	<Ctrl+C>	Copies the selected object to the clipboard.
	Paste	<Ctrl+V>	Pastes the object from the clipboard at the selected position.
	Delete	<Del>	Deletes the selected object and all following objects connected to the object.
	Rename	F2	A name is renamed using this function.
	Find	<Ctrl+F>	The "Search & replace" dialog opens. Various search functions are available.

Function			Description
	Zoom	25 % ... 300 %	**Only in the topology graph** Allows the size of the display to either be increased or decreased (zoom in / zoom out) in the topology graph. You can also freely change the size by holding down "Ctrl" and turning the mouse wheel.
	Reset graphic		**Only in the topology graph** Resets the topology graph to its standard representation.
	Select module from catalog		Selects the module based on its device class and the specified article number in the "module catalog".

10.3.2 Module catalog

Modules

The modules are saved with article numbers and a description in the module catalog. It is possible to insert the components specifically or unspecifically. Inserting is implemented by means of drag-and-drop or directly in the topology.

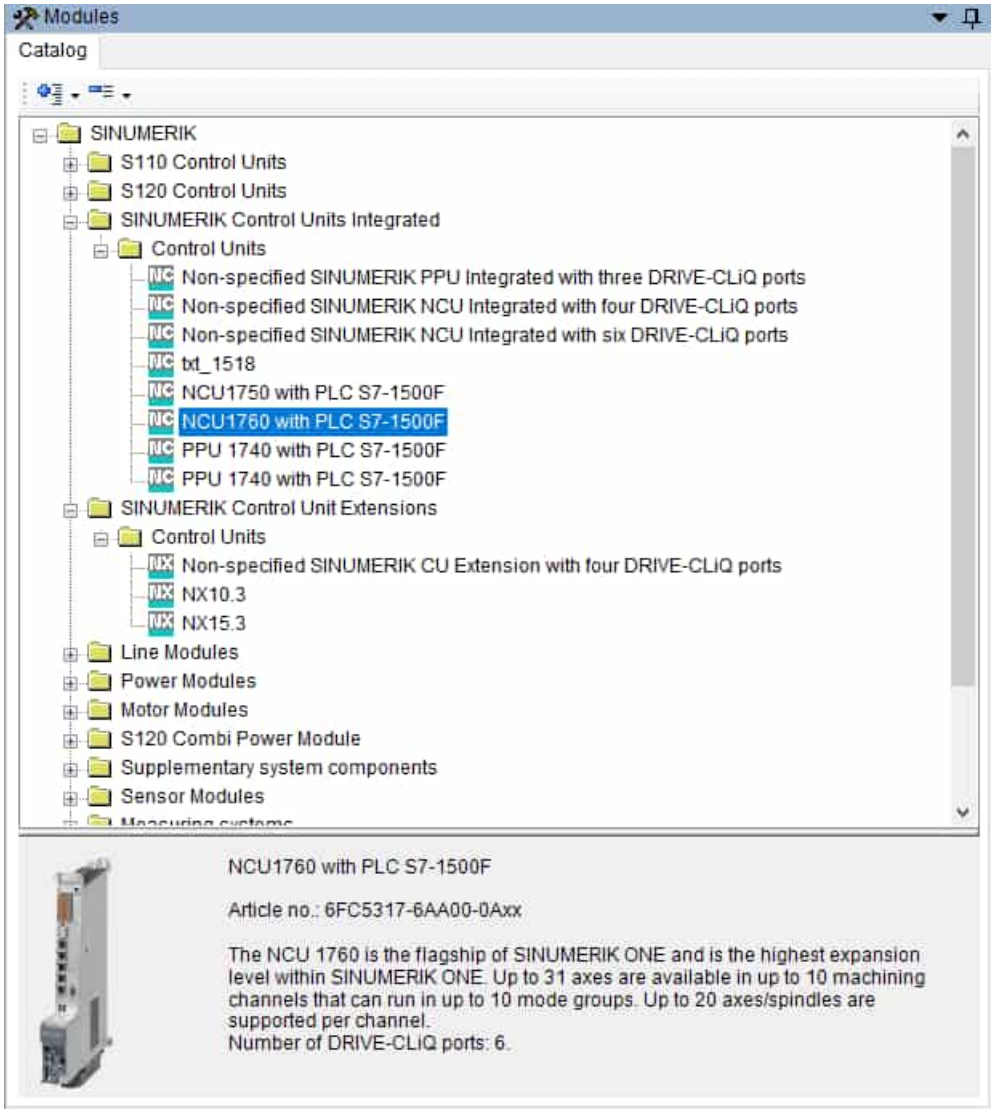
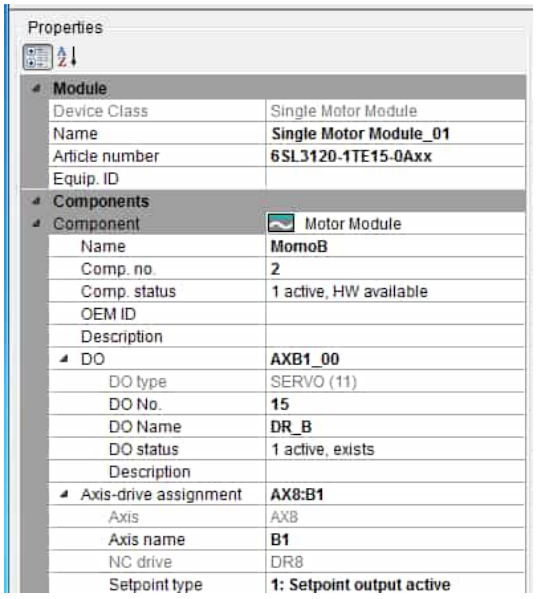


Figure 10-5 Module catalog

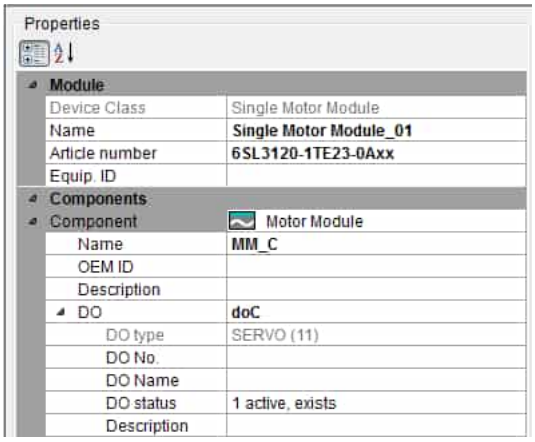
Component properties

Table 10-10 Each component can be assigned various properties, depending on the device class:

Component properties of the user-specified topology "*.utz2"



Component properties of the comparison topology "*.ust"



Property	Description	Note
Module		
Device class	Device class display	
Name	Unique name of the module	The module name is used only for the documentation and is not transferred to the control.
Article No.	Article number of the module	The package checks not only the wiring, but whether the correct module is used based on the article number. If a position in the article number should not be checked, you can do this by specifying an 'x', e.g. 6SL3120-1TE23-0Axx. If an article number differs for the package execution, the package generates an alarm that the user can ignore. If a "?" is in a position, this position is instantiated by the SINAMICS when reading in a user-specified topology.
Equip. ID	Equipment identifier	The equipment identifier is used only for the documentation and is not transferred to the control.
Components		

Property		Description	Note
Component		Display the used component.	
	Name	Name of the component.	Component names must be unique and have a maximum of 50 ASCII characters.
	Comp. no.	The CU-related SINAMICS component number is displayed or specified.	The component number can be edited. The following rules apply: • The SINAMICS component number range lies between "1" and "199". • Component number "1" is assigned automatically for Control Units and cannot be edited. • The component number must be unique for a Control Unit. • For all other components, the component number can be in the range from 2 to 193 for a control unit (CU320, NCU or NX) (6 C numbers are reserved for CU-LINK DOs). • In the topology configurator, the component numbers can be automatically reassigned or added via the shortcut menu when selecting a bus or slave.
	Comp. status	Setting options: • HW available, inactive (p1x5=0) • HW available, active (p1x5=1) • HW not available, inactive (p1x5=2)	• Default value is "1 active, HW available". • Cannot be changed for CU320 and NCU control units. • Also cannot be changed for Drive Objects (DO) that only comprise one component.
	OEM ID	ID of component in the catalog of the machine manufacturer.	Only numbers can be used.
	Description	Free description of the component.	The description is used only for the documentation and is not transferred to the control.
	DO		

Property		Description	Note
	DO type	Assignment of a Drive Object to a component type.	It is a property of the DO variable, and can be modified only using the "DO variable list (Page 439)".
	DO no.	DO number via which a drive object is addressed.	It is a property of the DO variable, and can be modified only using the "DO variable list (Page 439)".
	DO name	Freely selectable name for a drive object.	It is a property of the DO variable, and can be modified only using the "DO variable list (Page 439)". DO names can have a maximum of 24 ASCII characters.
	DO status	Setting options: • 0 inactive, available • 1 active, available • 2 not available	• Default value is "1 active, available". • For DO types "Integrated (NCU)" and "SINAMICS S", the status "active" cannot be changed.
	Description	Description of the drive object (DO).	-
Axis-drive assignment		Select the axis-drive assignment via this button.	-
	Axis	Display of the axis type, cannot be edited	-
	Axis name	Name of the axis.	-
	NC drive	Display of the NC drive.	-
	Setpoint type	Selection of the setpoint type.	-
DO variable		The specified DO variable is assigned to this component. The drop-down list lists all DO variables available in the DO variable list that are permitted on this component in accordance with their type.	Does not exist for device classes that do not constitute a drive object, e.g. motor, encoder. A DO variable name must start with a letter or an underscore and comprises the characters A-Z_0-9.
Encoder number		Specifies the number of the Encoder Data Set (EDS) to which this encoder evaluation with its associated encoder data set is assigned in the referenced DO.	Only permissible in conjunction with a DO variable reference. Only for the "Sensor Module" device class.

10.3.3 DOList Editor

To edit a DO list independently of Expert and Topo, use the "Create MyConfig DOList Editor". Using the "DOList Editor", you can create and update a DO list saved centrally in the company.

Open the "DOList Editor" from the Windows start menu

The "DOList Editor" is inserted in the Windows start menu when installing CMC. If you want to create a new DO list, you can directly start the "DOList Editor" from the Windows start menu via the start menu item "DOList Editor 6.3".

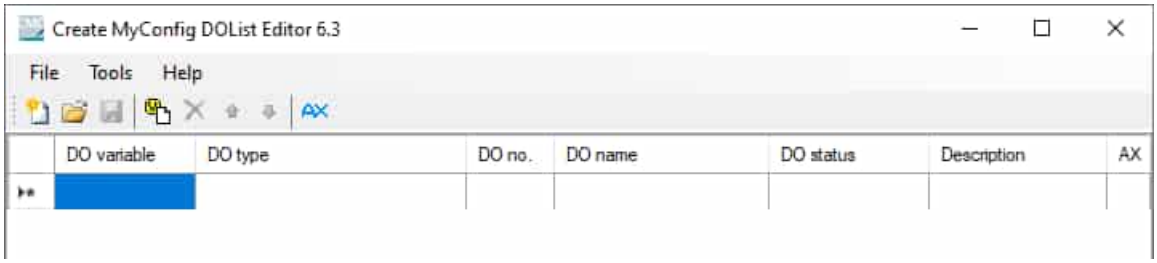


Figure 10-6 DOList Editor

Main menus and toolbars

The following main menus and tool bars are available for the "DOList Editor".

Table 10-11 File main menu

File			
	New	<Ctrl+N>	Creates a new DO list.
	Open	<Ctrl+O>	Opens an existing "*.uvdx" DO list.
	Save	<Ctrl+S>	Save the current DO list with file extension "*.uvdx". If components are changed, they are marked with an asterisk at the component. The diskette icon is active so that the changes can be saved.
	Save as...		Saves the current DO list with a different path or file name.
	Properties...		Displays the properties of the DO list.
	Last opened		The DO lists opened last are displayed with path and file name. You can select the DO list directly here.
	Exit	<Alt+F4>	Exits the "DOList Editor".

Table 10-12 Tools main menu

Tools		
	Options...	Opens the "Options" dialog > "Language" tab. You can switch between the German and English user interface languages here. The language change only becomes active the next time that the "DOList Editor" is called.

Table 10-13 Help main menu

Help			
	Content...	<F1>	Opens the online help for the "DOList Editor". In the open online help, navigate directly to the required topics for Topo.
	About...		Contains information on the version and copyright.

Buttons for the "DOList Editor"

Table 10-14 Toolbars

	**New** Creates a new DO list.
	**Open...** Opens an existing "*.uvdx" DO list.
	**Save** Saves the current DO list as a DO list with the "*.uvdx" file extension. If components are changed, they are marked with an asterisk at the component. The diskette icon is active so that the changes can be saved.
	**Add DO** The components for the displayed steps are displayed or hidden via this button.
	**Delete** Deletes the selected components and all following components connected to the component.
	**Move up/Move down** Using the arrow keys, you navigate forward and backward between the DO lines or component lines.
	**Predefinition of axis assignment** DO, which is a marked with icon , a predefinition of axis assignment is also configured. Using this button, you open the configuration table "Predefinition of axis assignment".

DO list in the "DOList Editor"

Display of the DO list in the "DOList Editor":

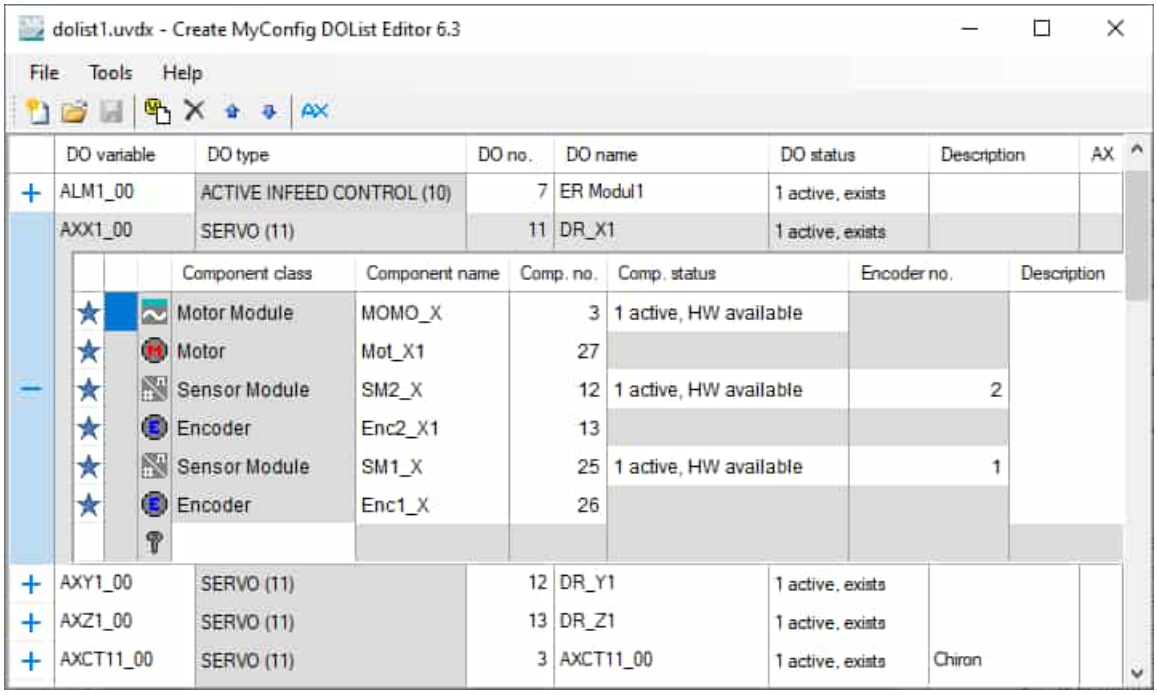


Figure 10-7 View of the DO list in the "DOList Editor"

10.3.4 Manage DO list

Call the DO list via the main menu "Data" > "Manage DO list".

Using the DO list, manage the DO variables used in the Topo project.

You can manually create DO variables, or import from an existing UVDX file and edit. The DO list serves as interface between the Expert and Topo configuring tools.

Topo exports DO variables into a DO list "*.uvdx" or imports from this . You can also link a DO list .

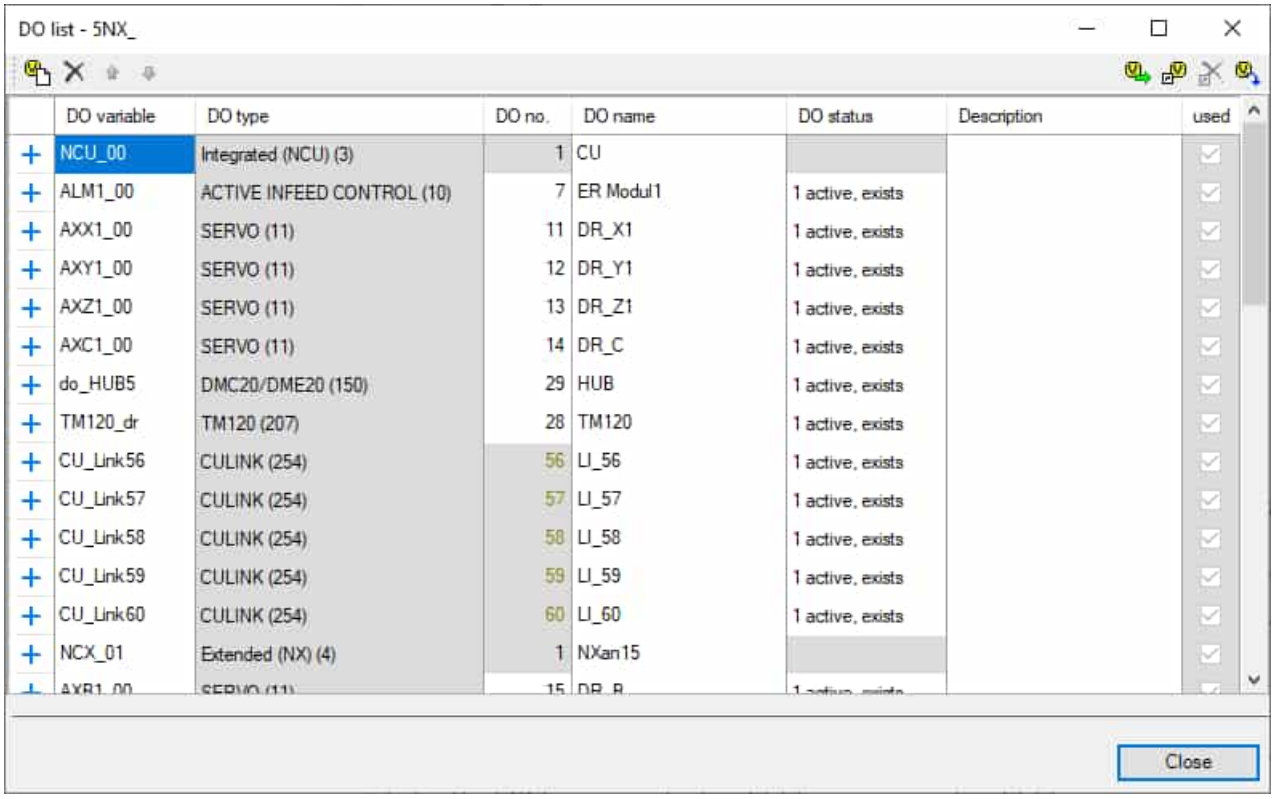


Figure 10-8 Example of a DO list

DO variables - assign properties

Table 10-15 You can assign the following properties to any DO variable:

Property	Description	Note
DO variable	Name of the DO variable used in the package for assignment to a drive object.	Valid characters: a-z, A-Z, 1-9, _ Please note: • No number at the front • Min. 2 characters • Max. 24 characters
DO type	Defining the DO type.	-
DO no.	DO number via which a drive object is addressed.	The following numbers are permitted: • 2 to 54 • 61 and 62 Permissible for control units (CU_I, NX): • 1 For DO numbers: Is reserved for independent BICO connections within a DO: • 63 Reserved for "CU-LINK"- DO: • 55 - 60

Property	Description	Note
DO name	Freely selectable name for a drive object.	Max. 25 characters
DO status	Set the DO status.	Setting options: 0 inactive, available 1 active, available 2 not available
Description	Freely selectable description text for the DO variable.	-
used	The drive object (DO) is only used in the topology if the option box is activated.	Display only

DO type

DO type	Module assignment
SINAMICS S Integrated (NCU) Extended (NX)	Control Unit
Determine INFEED from HW ACTIVE INFEED CONTROL SMART INFEED CONTROL BASIC INFEED CONTROL	Line Module
Servo (11) - servo, servo HF Servo HLA (70) - servo HLA	Servo, Servo HF
Determine TM from HW TM31 TM41 TM15 TM120	Terminal module
TB30	Terminal Board
DMC20 / DME20	Hub Module
CU-LINK	-

Assigning DO variables

In dialog "Properties", you assign DO-forming and DO-referencing components to a DO variable.

The properties of the DO variables assigned to a component are displayed in the "Properties" dialog box.

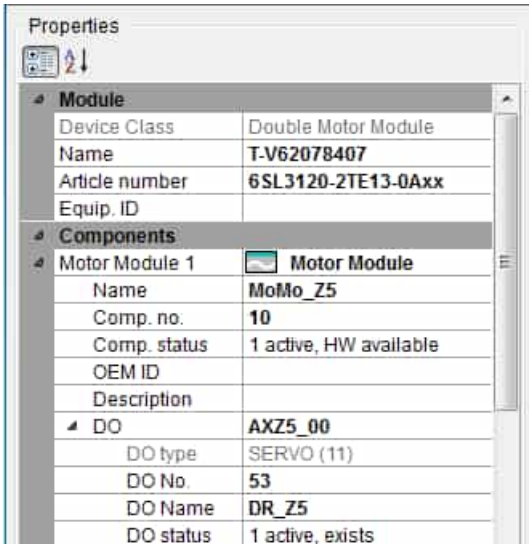


Figure 10-9 Component properties

10.3.4.1 Local DO list

Each topology of a project has a separate variable table for local DO lists.

If the same variables are configured for different topologies or projects, link them to an external list.

When importing archives, DO variables are automatically formed from the DO names. When commissioning with Create MyConfig, the DO variables are stored in SINAMICS and are therefore available again for the archive import.

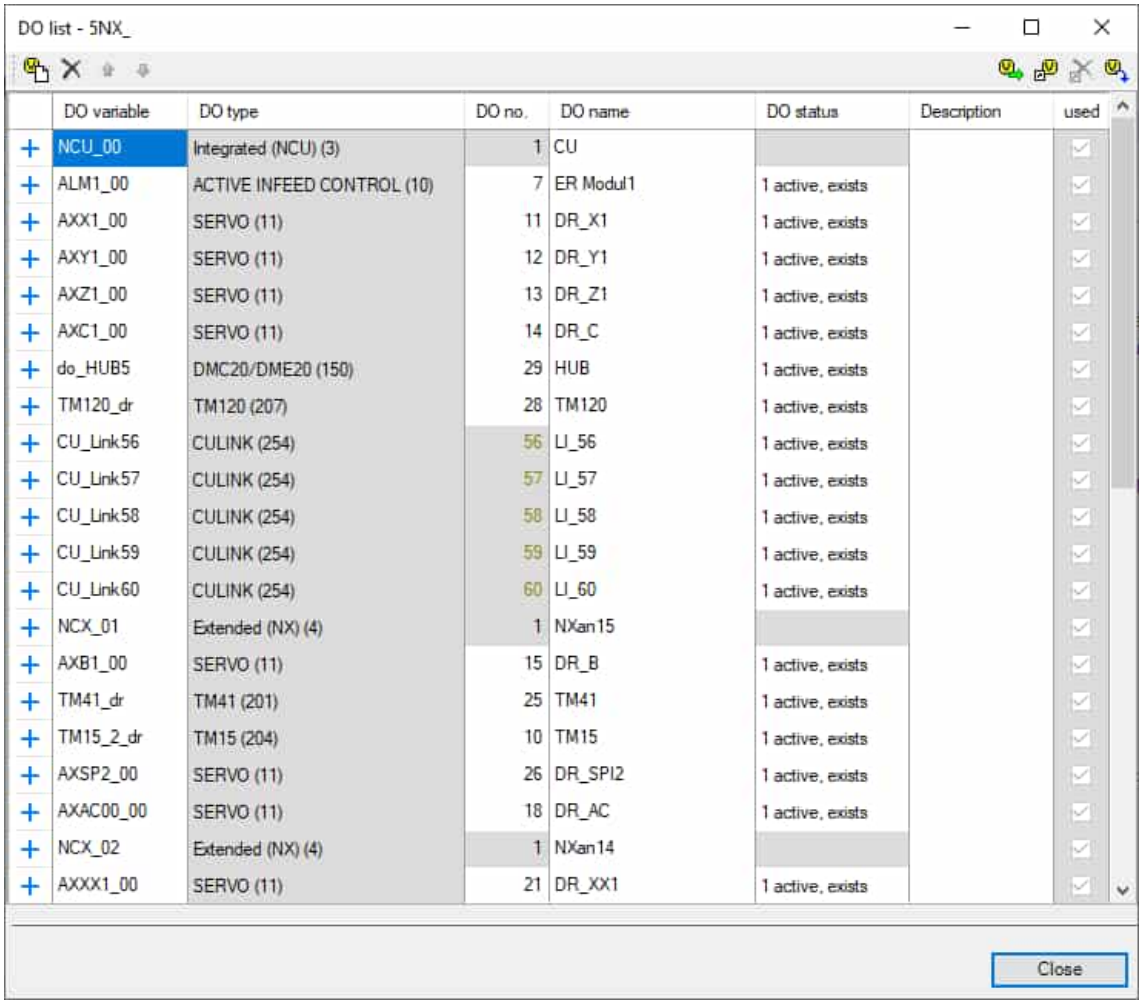
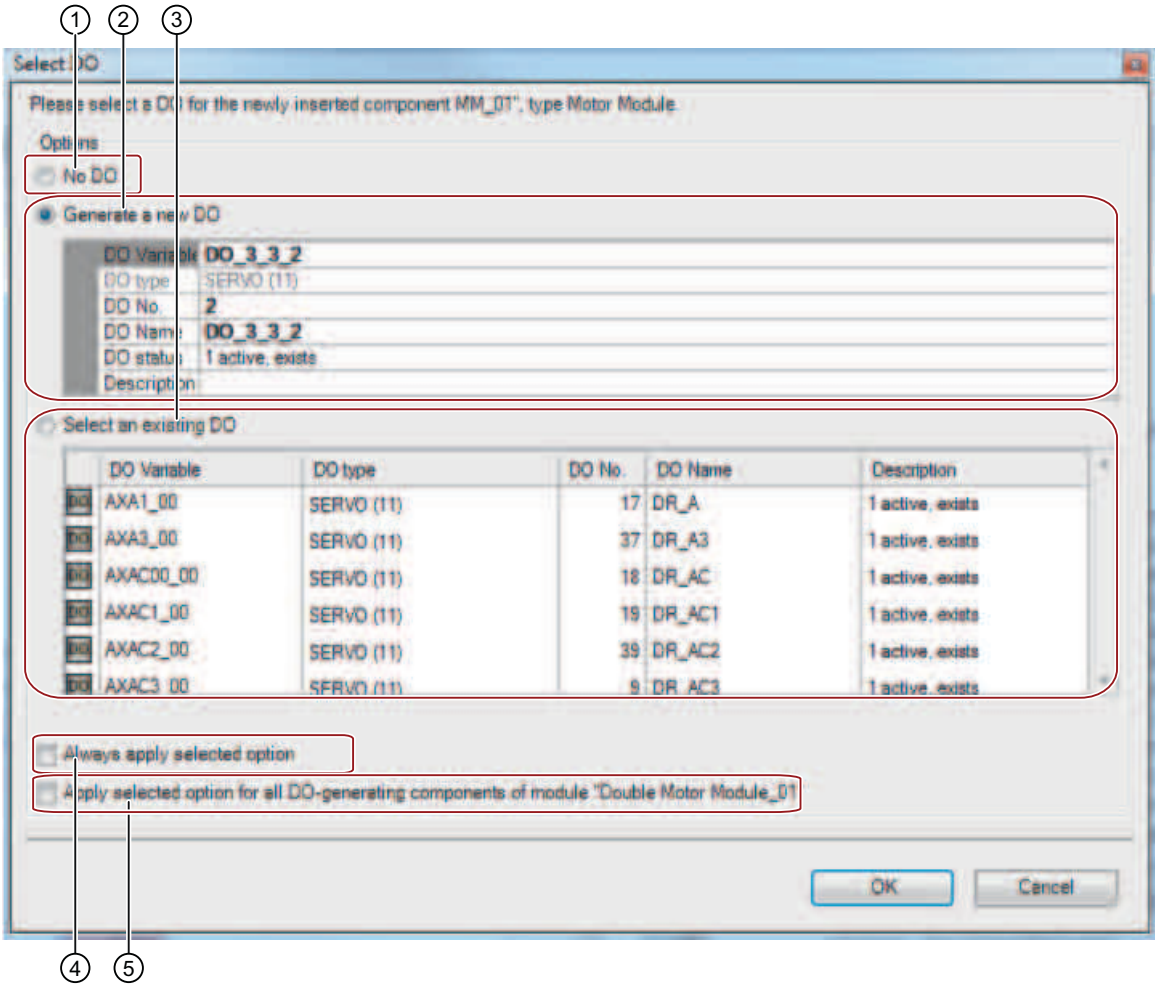


Figure 10-10 Local DO list - displays of linked and unlinked DO variables

10.3.4.2 Generating DO variables automatically

If a new DO-forming component is added to a topology, a DO variable can be generated.



① No DO is created for this selection.

② A DO is generated. DO variable, DO number and DO name can be edited. The DO type is determined automatically.

③ An existing DO is selected from the list.

④ If the "Always apply selected option" option is activated, this dialog is not redisplayed when components are added. The selected option is always used without a new prompt.

Exception:

This action is hidden for option ③ "Select an existing DO".

⑤ Option "Apply selected option for all DO-generating components of module" "<module name>" is only displayed for several components in a module (DMM, Combi, HLA, etc.). If this option is activated, the selected option is also repeated for the other components of the module.

Exception:

This action is hidden for option ③ "Select an existing DO".

Figure 10-11 Generating DO variables automatically

10.3.4.3 DO list

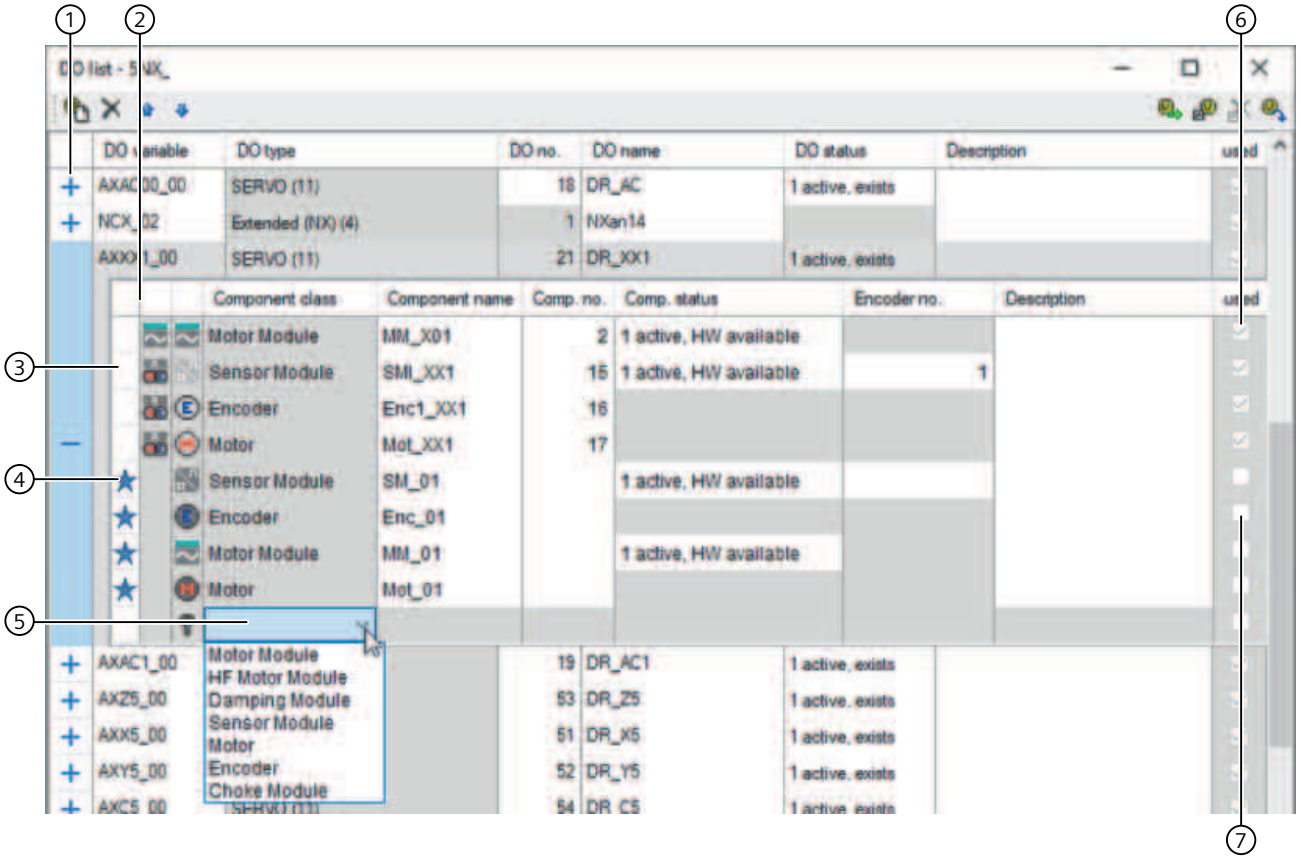
To preconfigure an assignment of components to certain drives (DO), the DO in the "DO list" can be extended to include components.

The option of permanently configuring components to certain drives (DO) is used in the following scenarios:

1. **Use in Topo as basis for configuring topologies**
   A DO list can be configured and exported in Topo. The DO list is saved centrally in the company, for example, for a machine series.
   Previously, you manually entered and assigned the component properties and DO properties when creating new topologies. Alternatively, you can now use a DO list as basis for configuring. The "*.uvdx" DO list is imported or referenced from any storage location.
   The DOs and components configured in the DO list are offered for use when creating the topology.
   Only preconfigured drives and components are used for a referenced DO list.
2. **Use for configuring the EPLAN with the DQ topology export**
   As described in item 1, the configured DO list in Topo is used in the Siemens DQ topology export (EPLAN Add-In) for deriving a topology.
   Where the individual modules in EPLAN are assigned drives (DO) and components from the DO list.
   A description of the Siemens DQ topology export can be found in the "SINUMERIK Create MyConfig - EPLAN Electric P8 Add-In (https://support.industry.siemens.com/cs/ww/en/view/109769191)" documentation.

Editing the component list

The interface of the component list is described below.



① Open DO
Clicking the blue plus character opens a drive object (DO).
A new dialog displays the components that belong to the DO.

② Component display of a DO.
The component properties are changed in the components dialog.
Components can be added and/or deleted.

③ DO-independent components
These components are not permanently linked with the DO.
For example, these components are not kept when components are deleted.

④ Preconfigured components ★
Preconfigured components are obtained when components are newly created below a DO.
These components are not permanently linked with the DO and used only together with the DO.
The preconfigured components are also automatically copied when a DO is copied.
When a module in the topology is deleted, the preconfigured components used are not deleted.
Preconfigured components are deleted, copied and edited via the shortcut menu in the components dialog.

⑤ Insert a new component via the drop-down list.
Select a component class via the drop-down list. A component name is created automatically.

⑥ If the checkbox is activated, this indicates that the component is used in the topology.

⑦ If the checkbox is not activated, then this indicates that the component is not used in the topology.

Figure 10-12 Editing the component list

Exporting a DO list

When exporting a "*.uvdx" DO list, **all** components linked with the drives (DOs) are exported, irrespective of whether the components were configured via the DO list or assigned from the topology to the DO.

Importing a DO list

When importing a "*.uvdx" DO list into an existing topology, all DO assignments are deleted.

Linked DO list

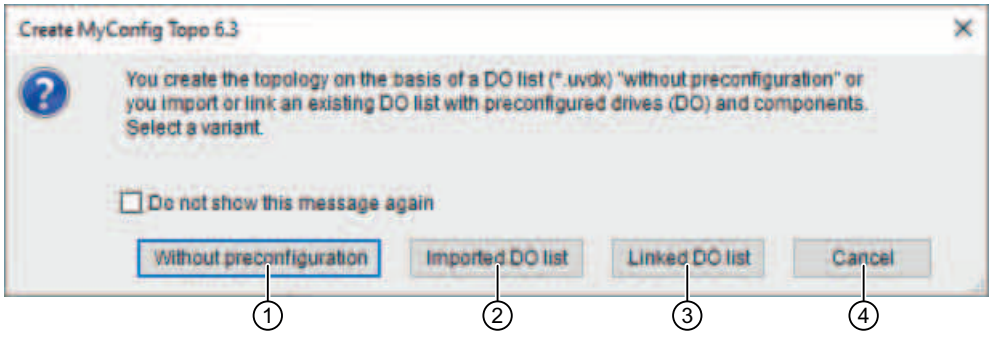
When linking a DO list, all existing assignments in the topology are removed. The DO assignment to components is only possible when a component of the correct type, which has not been used on any other component of the topology, is available for the selected DO in the DO list.

You can edit and add new DO and components in the linked DO list. This is possible only within the DO list by clicking the button.

After editing, save the DO list to apply the changes.

Using a DO list when creating a new topology

The following variants are available for creating a new topology:



①"Without preconfiguration": Opening an empty topology.

②"Imported DO list": An empty topology with an imported "*.uvdx" DO list opens.

③"Linked DO list": An empty topology with a linked "*.uvdx" DO list opens.

④"Cancel": The operation is canceled.

Figure 10-13 Creating the topology - using the DO list

If one of the versions based on a DO list is selected, the DO list contains preconfigured DO properties and component properties. A DO list simplifies the assignment of properties when configuring the topology.

Wiring modules

When wiring modules from the module catalog, the immediate assignment of a DO is initially possible for each DO-forming or DO-referencing component of the module per dialog.

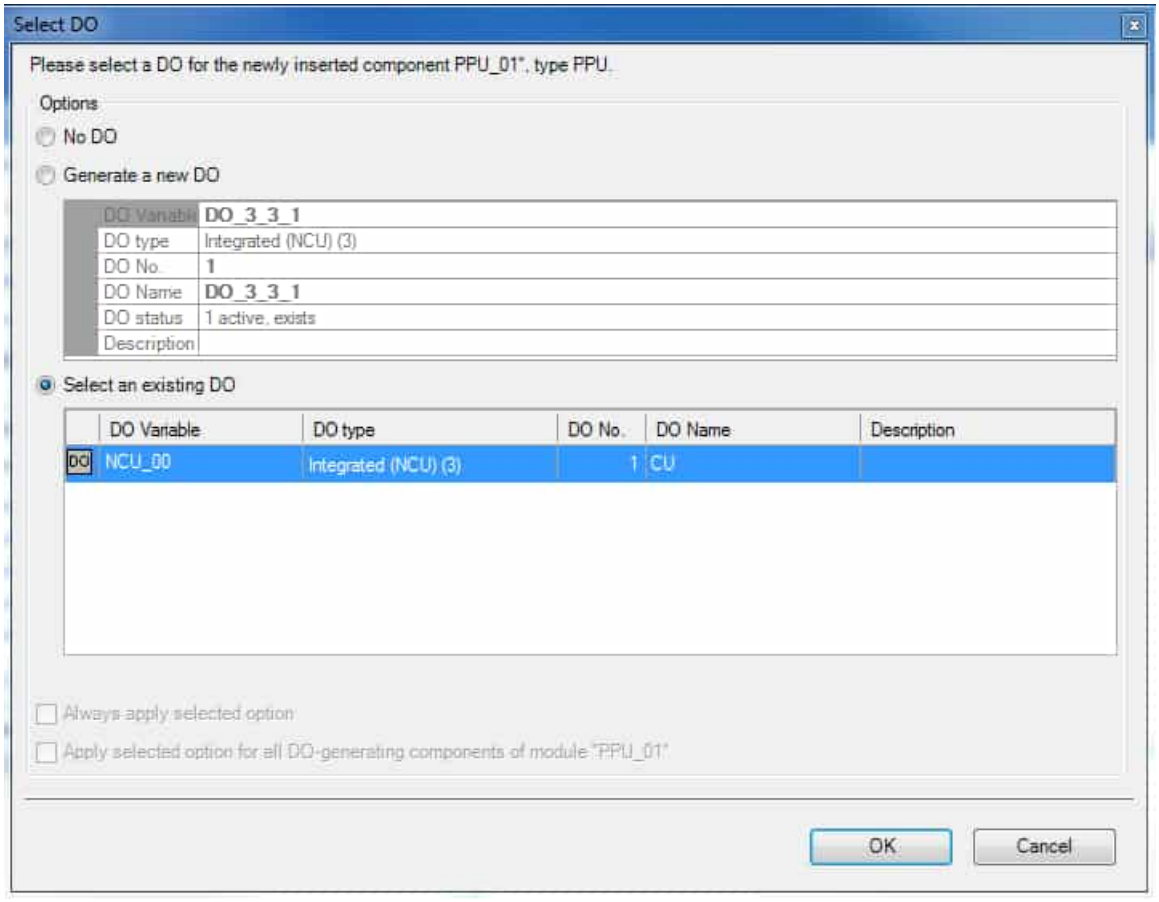


Figure 10-14 "Select DO" dialog

If an existing DO is selected from the DO list, a preconfigured component can be selected from the component list for the selected DO in a further dialog for each component of the module.

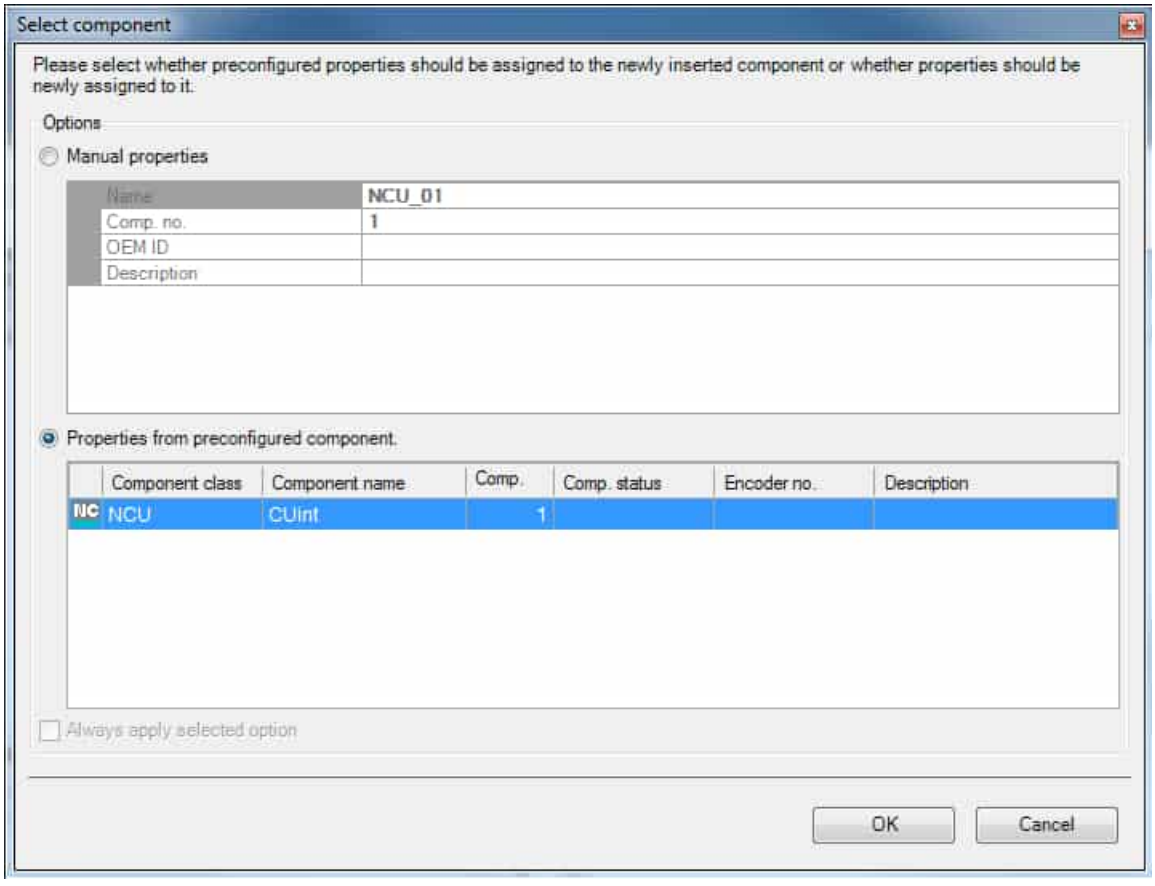


Figure 10-15 "Select component" dialog

Note

Linked DO lists

Only preconfigured components are available for linked DO lists.

Use of a DO list on an existing topology

If you import a "*.uvdx" DO list into an existing topology, then initially, all DO assignments in the topology are deleted.

For new assignments of the DOs via the component properties, the preconfigured components are listed for use in the DO list.

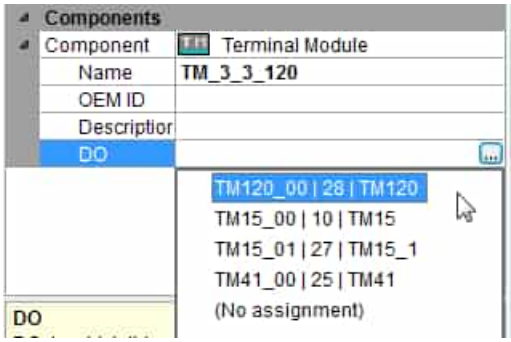


Figure 10-16 DO list - preconfigured components

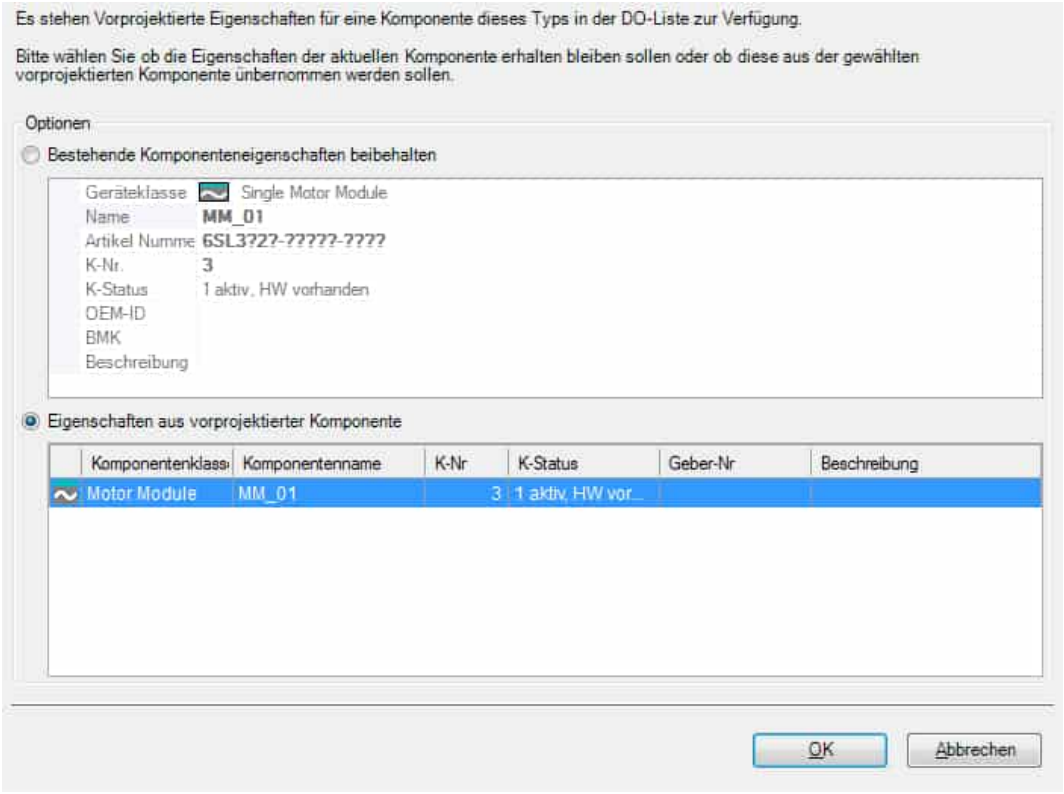


Figure 10-17 Preconfigured properties for a component

The following options are available:

- "Retain existing component properties"
  The existing component properties are retained or can be adapted directly in the dialog.
- "Properties from preconfigured component"
  The component properties from the preconfigured component of the DO are used.
  If this option is selected and confirmed with "Yes", the component is added as "preconfigured component" to the component list.
  If this option is selected, but rejected with "No" in the following dialog, the component is removed.

10.3.4.4 Predefinition of axis assignment for the EPLAN add-in

You can use the table in the "Predefinition of axis assignment" dialog to configure the required assignment to the axis for each DO in the DO list.

The predefinition of axis assignment is only used for the import of a topology from the EPLAN Add-In via the Siemens Data Exchange "*.sdx" format.

Whereby, the preconfiguration of the DOs used in the specific topology is performed in the axis-drive assignment.

You can specify an axis name, the setpoint type, as well as the NC drive to be used for the transport, for a selected axis.

In addition to the setpoint interface, you can also specify the axis to which the two encoders are to be assigned. You can also configure the actual value type to be used and the NC measuring system.

The table allows any number of DOs to be assigned to the same axis. If the topology selected for the import in Topo uses several DOs, for which the assignment to the same axis has been configured, this assignment is rejected with a message.

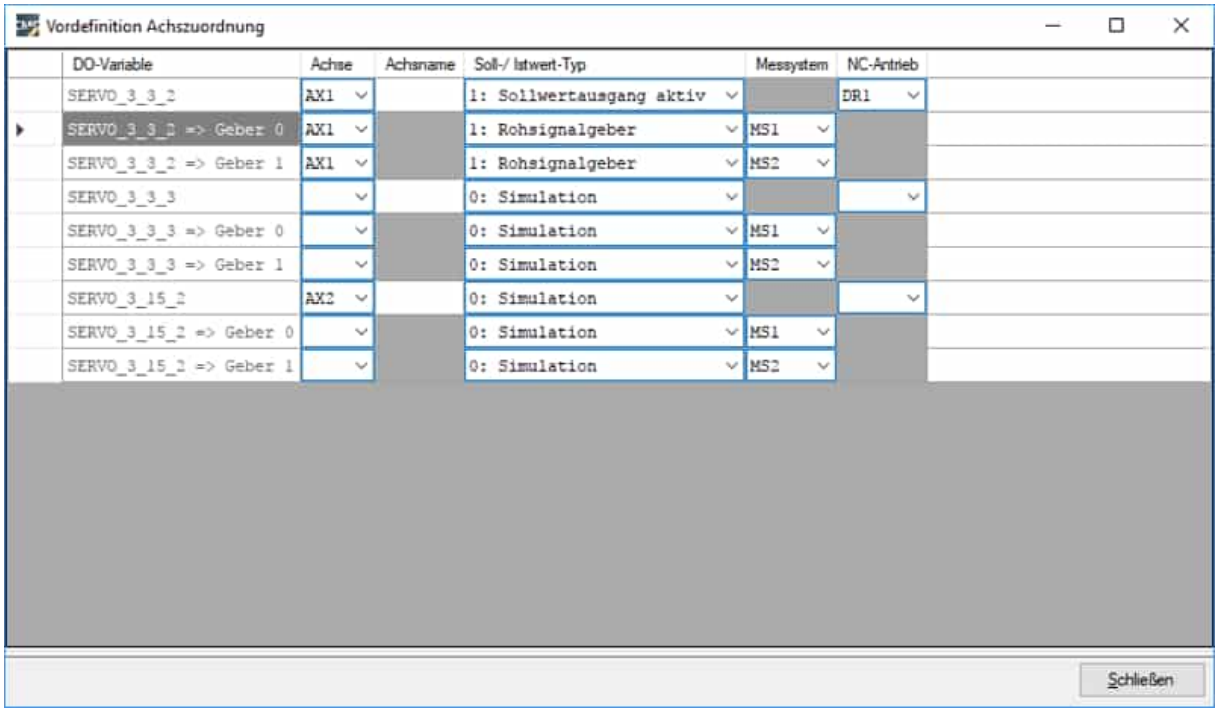


Figure 10-18 Table in the "Predefinition of axis assignment" dialog

10.3.5 Topology comparison

The topology comparison provides you with the option of comparing 2 topologies as well as components - or to synchronize the individual properties of a component.

Using menu "Comparison" > "Refresh" , you can compare 2 topologies contained and marked in the current project.

Using menu "Comparison" > "Compare with external topology", you can compare a topology contained in the project with a topology from an archive or a "UST" or "UTZ2" file.

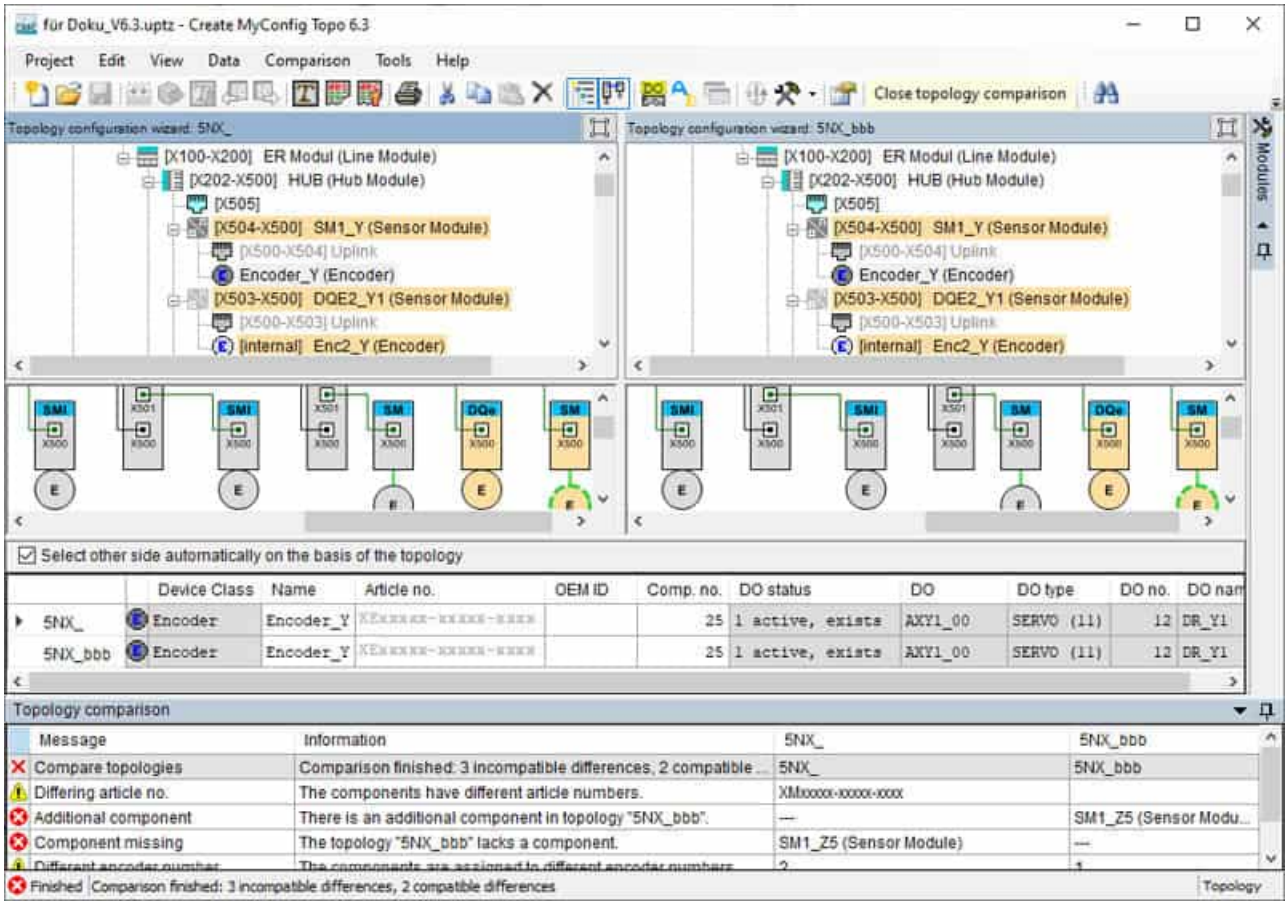


Figure 10-19 Comparing topologies

Marking the differences in the component properties

The Properties table displays differences in the properties of the components and at the same time offers the option of editing the properties. In addition, it is also possible to synchronize individual properties of the component or also complete components between two topologies via the shortcut menu. Differences in the properties of the two components that have been selected are shown in color.
For the article number, differences at the individual positions are also shown in color.
The colors have the following meanings:

Red	Property is available in both components, but is different
Blue	Property is only available in the component selected in the right-hand topology

Green	Property is only available in the component selected in the left-hand topology
Gray	Digit of the article number irrelevant (x)

Identifying the differences in the topologies

After the topologies have been compared, then the result of the comparison is shown in the lower section. You can jump directly to the location in the topologies where the differences were identified via the shortcut menu "Go to difference" or by double-clicking the message line.

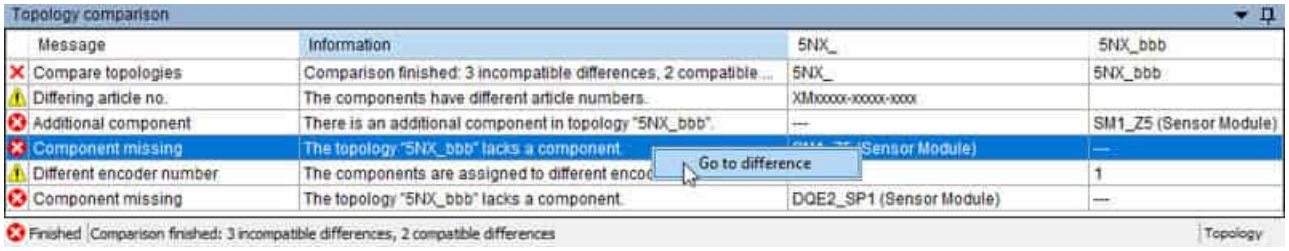


Figure 10-20 Topology comparison - messages

10.3.6 Forwarding a topology

Procedure

To deploy the topology in Topo, proceed as follows:

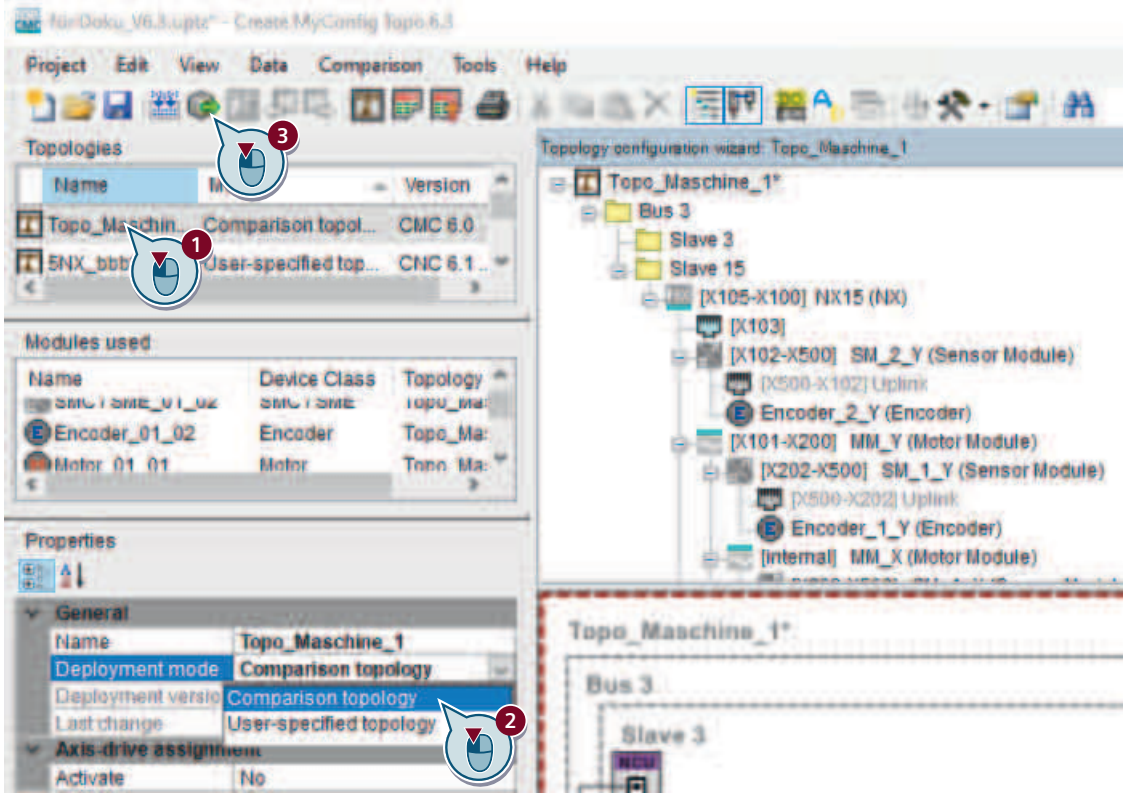


Figure 10-21 Forwarding a topology

1. Select the topology that you want to forward.
2. Set the Deployment mode of the topology to deploy this as "*.utz2" user-specified topology or "*.ust" comparison topology.
3. Start the deployment.

10.4 EPLAN Add-In

10.4.1 Inclusion of DRIVE-CLiQ topologies from EPLAN electric P8

An Add-In for "EPLAN electric P8" for exporting the structure of the DRIVE-CLiQ topology from a circuit diagram is included in the scope of delivery of Create MyConfig. In Create MyConfig, the topology files exported from a circuit diagram can be used to fill a machine.

The exported topologies are processed in Create MyConfig using the "ECAD Import" extension, which is subject to license.

More information

You can more detailed information on the function:

- In this documentation
- In the SINUMERIK Create MyConfig – EPLAN Electric P8 Add-In (https://support.industry.siemens.com/cs/ww/en/view/109769191) Operating Manual

10.5 Command line call

10.5.1 Topo

Call

Write the commands shown in the following as well as the used switches and arguments in one line to a batch file or a command line interface.

The significance of the switches and arguments is described in the following "Topo parameter" table.

Starting Topo

Topo is started with the following command, and a file can be displayed and edited.

```
Topo <Topology path>
```

Comparing topologies

2 topologies can be compared with the following command, i.e. 2 files are displayed and can be edited.

```
Topo /diff <File1> <File2>
```

Automated Topo deployment

For automated deployment, you can execute Create MyConfig Topo in the command line mode.

```
Topo /silent <Topology path>
     /deploy [<Deployment name>] [/deploydir <Deployment directory>]
     /target (UST|UVDX|(UTZ2)) [/autocomplete]
     [/log <Log file>] [adduvdx <Uvdx path>] [extendcompnames]
```

The possible deployment modes are listed in the following table.

Table 10-16 Deployment modes

Deployment mode	Explanation
UST	Comparison topology
UTZ2	User-specified topology
UVDX	DO and components used in the topology as DO list.

Notes

Note the following information for the command line call:

Note

Deployment version

If no deployment version is specified for the "UTZ2" deployment mode, then deployment versions defined in the source for the individual CU are used. If no deployment version is defined, the newest version is deployed.

Note

Observe the sequence of the switches

Write the switches in the specified sequence.

When doing this, observe the "Call" section and the "Significance of the switches" table.

Note

Path data

- Set the specified paths in straight quotation marks if spaces are included.
- **Example**:
  "C:\Path with spaces\Expert_Logfile.log"
- Paths must never end with '\'.

Note

Saving batch files

Always save batch files with "ANSI/ASCII" coding.

Note

Umlauts and special characters

Avoid the use of umlauts and special characters in path names.

To allow special characters or umlauts to be correctly processed in specified paths, save the batch files with command line calls with the same character set as used by the region-specific command line interpreter (cmd.exe).

For example, Codepage "850", also known as "DOS Latin 1", applies for region Western Europe.

The "Notepad" editor integrated in Windows does not support saving files with this character set.

Note

Uppercase/lowercase

For the command line switches and arguments, **always** use the specified uppercase/lowercase notation.

Note

Absolute/relative path

Either the absolute path or the relative path can be specified.

The relative path with ".\" or "./" refers to the current work folder, where the batch or CMD was called.

If no path is specified, the work folder also applies.

Parameter

Significance of the switches

Table 10-17 Topo - parameters

Switch	Arguments	Optional	Description
-	<Topology path>	No	• Specifies the path of the project file – either absolute or relative. • Set the specified paths in straight quotation marks if spaces are included. **Example**: "C:\Path with spaces\package file" • For topology projects "*.uptz", the name of the topology to be compared must be specified separately after the path with a colon. If a name is not specified, then the first topology contained in the project is selected. • The following are permitted: "*.uptz", "*.ust", "*.utz2", "*.sdx", "*.uvdx", as well as xml files according to the topology interface description and SINUMERIK archives
/diff	<file1> <file2>	No	• 2 paths to "*.uptz", "*.ust", "*.utz2", "*.sdx", "*.uvdx", arc files • For "*.uptz" topology projects, the name of the topology to be compared must be specified after the path separated by a colon.
/silent	-	No	• The graphical user interface of Create MyConfig Topo is hidden. • The following options are only effective with this switch.
/deploy	<Deployment name>	No	• Causes the topology selected in the project to be checked and deployed. • If the parameter is not specified, Topo generates the name from <Topology path>.
/deploydir	<Deployment directory >	Yes	• Specifies the deployment path, absolute or relative, in which the topology is to be deployed. • Set the specified path in straight quotation marks if spaces are included. **Example**: "C:\Path with spaces" • If the switch is not specified, Topo uses the <Topology path> folder.

Switch	Arguments	Optional	Description
/target	(UST|UVDX|(UTZ2[60]))	No	• Deployment type and CNC software version of the NCU, on which the user-specified topology is applied. **Note** **Deployment version** If no deployment version is specified for deployment mode "UTZ2", then the deployment versions defined in the source at the individual CU are used. If a deployment version is not specified in the source, then the latest version is deployed.
/autocomplete	-	Yes	• Automatically adds missing component numbers and DO properties and sets up standard encoder assignments if none have been specified.
/log	<Log file>	Yes	• Specifies the path of a log file in either absolute or relative terms. • Alarms and faults are written to the log file (XML format) when checking and deploying the package. • Specification of this switch with its parameter is optional.
/adduvdx	<Uvdx path>	Yes	• Specifies the path to an additional UVDX file when importing an SDX file. • If the SDX does not contain any DO and component information in the form of a UVDX, this is added based on the references contained in the SDX from the specified UVDX. • If the SDX file already contains DO and component information, the action is canceled with an error message. • Is only effective when an SDX file has been selected as topology path.
/extendcompnames	-	Yes	• When importing a topology from an SDX file, you can also select whether the configured component names should be supplemented with the equipment identifier (Equip. ID) specified for modules in EPLAN. • The resulting component name corresponds to the following scheme: `<equipment identifier>; <component name>` • Is only effective when an SDX file has been selected as topology path.

Return values

Note

The return values for the Topo command line call apply only if Topo is executed in `/silent` mode without user interface.

The return values from Topo can be determined by querying the `%ERRORLEVEL%` variables. More information can be found in the following example.

Table 10-18 Topo - return values

Value	Description
0	No faults or alarms occurred when checking/deploying the project.
1	Alarms but no faults occurred when checking/deploying the project.
2	Faults (and possibly alarms) occurred when checking/deploying the project.
3	Command line call error.

Example

Precondition: Create MyConfig is installed in the "C:\Program Files\SIEMENS\Create MyConfig 6.3" directory and a "TopoProj_1.uptz" project that contains the "Topo1" and "Topo2" topologies is available in path " C:\Topology".

```
@echo off
PATH=%PATH%;"C:\Program Files\SIEMENS\Create MyConfig 6.3"

Topo.exe /silent "C:\Topologies\TopoProj_1.uptz:Topo1" /deploy xyz /deploydir "D:" /target
UTZ2


if %ERRORLEVEL% EQU 3 (
@echo "Command line call error!"
pause
)
if %ERRORLEVEL% EQU 2 (
    @echo "Error during transmission!"
    pause
)
if %ERRORLEVEL% EQU 1 (
    @echo "Warnings during deployment!"
    pause
)
if %ERRORLEVEL% EQU 0 (
    @echo "Deployment performed without errors!"
    pause
)
```

10.6 SINAMICS device configuration

10.6.1 Production of modular machines

Engineering software Create MyConfig (CMC) supports both the automated production of modular machines for machine manufacturers as well as the upgrade of machines for end customers with only a limited number of operator actions required.

Producing machines with SINUMERIK ONE requires basic commissioning of the SINAMICS, which is generally designated as "System configuration". You can configure the system using CMC, either as "Automatic configuration" or as "User-specified topology".

When you configure the system in the CMC package, in the step tree you can specify all the data sets and optimization data that are required within a series of modular machines in the maximum machine configuration.

The package is adapted to the specific version of a modular machine either initially during the configuring or when the package is being executed on the machine – either manually or using configuration files.

10.6.2 SINAMICS device configuration with Automatic configuration or user-specified topology

Applications

With regard to the device configuration of the SINAMICS, a general distinction is made between 3 use cases that CMC supports:

1. SINAMICS was commissioned previously with SINUMERIK Operate or with different commissioning software. With a CMC package, via "Initial state" (online access) on dialog page "System configuration" under "DRV archive data/drive configuration", a check of the existing drive data including the topology can be carried out for the drive archive data. In addition, a further configuration or optimization of the SINAMICS can be made.
2. The maximum configuration of the SINAMICS is wired at the machine, but the SINAMICS has not yet been commissioned or configured. Using a CMC package, the "Automatic configuration" under "DRV archive data/drive configuration" is selected from the "System configuration" dialog page. The checks, configurations or optimizations listed in item 1 in the same package are then performed.
3. Not all the SINAMICS components provided for the maximum configuration of the machine have already been wired on the machine. For example, DQI/SMI motors are still missing. SINAMICS has not yet been commissioned and configured. Using a CMC package, the "User-specified topology" under "DRV archive data/drive configuration" is executed from the "System configuration" dialog page. The user-specified topology file is a SINAMICS offline topology package (SOT) that also contains the topology in the maximum configuration of the machine. The checks, configurations or optimizations listed in item 1 in the same package are then performed.

Differences between "Automatic configuration" and "User-specified topology"

The table below shows the major differences between "Automatic configuration" and "User-specified topology." In the same way as the automatic device configuration via SINUMERIK Operate, the PROFIBUS assignment (p978), the telegrams and the BICO links are also entered automatically in both CMC variants.

Table 10-19

Automatic configuration	User-specified topology
Execution via SINUMERIK Operate or CMC (identical result)	Execution only via Create MyConfig
SINAMICS performs an automatic configuration and determines its own actual topology.	A user-specified topology is transferred to the SINAMICS in the CMC package that is applied as the target topology.
Only those SINAMICS modules wired at the time of automatic configuration are detected.	The user-specified topology can also include SINAMICS modules that have not yet been wired and deactivated. Exceptions are: • NCU • NX • CU320
The DRV archive data also contains the DO and components that have been wired after the automatic configuration was performed.	After importing the user-specified topology, the DRV archive data contains all DOs and components that were specified in the user-specified topology.
The step-by-step commissioning is possible only to a limited extent (e.g. the addition of DQI/SMI motors).	The step-by-step commissioning is possible almost without limitation (e.g. the addition of DQI/SMI motors).
The specification of the comparison topology file allows the DRIVE-CLiQ wiring to be checked.	The SINAMICS signals an error if the topology specified in the user-specified topology cannot be implemented.

Configuration in Expert

When configuring in Expert, on dialog page "System configuration" in properties group "DRV archive data/drive configuration" under "Origin of data" you define which of the 4 applications "Initial state", "Automatic configuration", "User-specified topology" or "Offline archive" is applicable.

When selecting "User-specified topology", it is imperative to generate a file with the user-specified topology with Topo, and to make it available to the package for execution on the "SINAMICS topology" dialog page. The user-specified topology file has a binary format with the file extension "*.utz2".

If "Automatic configuration" is performed, then on dialog page "SINAMICS topology" you can optionally use a file with comparison topology in the XML format and file extension "*.ust" or "*.xml". This is used for comparison with the automatically detected topology.

The two topology files can be configured in one package. During execution, the "Origin of data" setting under "DRV archive data / drive configuration" on the "System configuration" page determines which of the two settings will be used on the following "SINAMICS topology" page.

Both the user-specified topology as well as the comparison topology contain the topology of the SINAMICS and additional properties of the DO and the components.

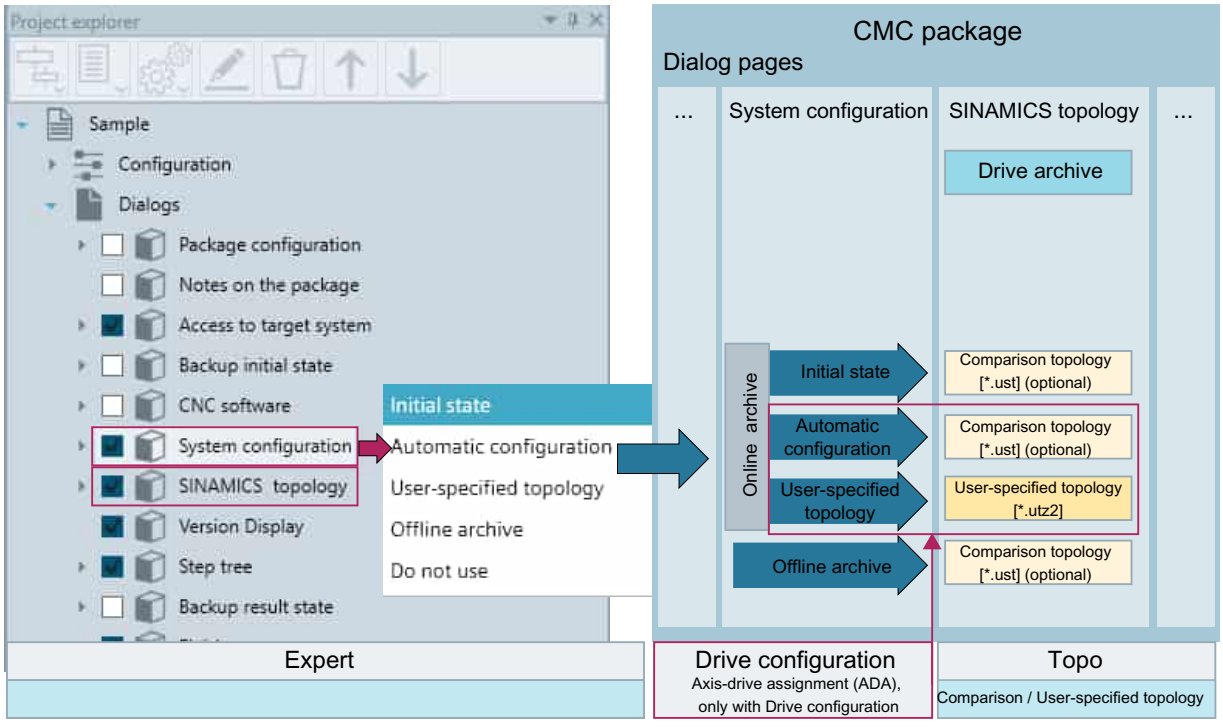


Figure 10-22 CMC package structure - SINUMERIK ONE

10.6.3 Drive data generation via "automatic configuration"

Basic procedure

The "Automatic configuration" via SINUMERIK Operate or via Create MyConfig uses the same functions internally. In both cases, SINAMICS determines the "actual topology" and saves this as "target topology" (p9903). This also includes the creation of drive objects (DO) and their assignment in a folder structure according to bus number and slave number. Each DO is represented as a PS file either in ACX or ASCII format "*.TEA". File "PS000000" is used to monitor the consistency and file "PS000099" contains configuration data of the internal PROFIBUS. This structure can be saved as drive archive and imported later.

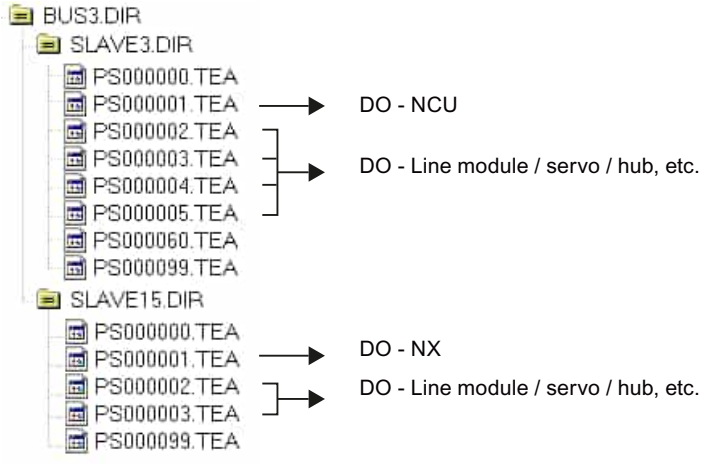


Figure 10-23 Drive data storage

For the "Automatic configuration", SINAMICS assigns the following:

- DO names
- DO numbers
- Component names
- Component numbers
- Component properties:
  - Article number
  - Component category
  - Component type

The parameters for the DO are read-out from the components, calculated or assigned the factory settings.

The complete data is backed up in the drive data archive.

10.6.4 Drive data generation via "user-specified topology"

Basic procedure

If the SINAMICS device is configured using the "user-specified topology", then the CMC package specifies a SINAMICS offline_topology package for the SINAMICS in the form of a user-specified topology file "*.utz2", which corresponds to the maximum machine configuration. In other words, components, for example, "DQI/SMI motors", still may be missing because commissioning has been performed step-by-step.

In addition to the topology, the user-specified topology file also specifies the following data:

- DO names
- DO numbers
- Component names
- Component numbers
- Component properties
  - Article number
  - Component type

The structure with generalized drive data is also specified.

Finally, when executed, the package requests that SINAMICS checks this data based on the currently available configuration and customizes it appropriately, i.e. define the DO or component status.

As a result SINAMICS has a customized target topology (p9903) and customized DO for the specified components.

The DO parameters are customized by SINAMICS with the factory settings, exported from the components or calculated. The complete data is backed up in the drive data archive.

Because at this time components not yet present in the structure have already been created, you can add the associated user data to the step tree using scripts. As an alternative, for example with DQI/SMI motors, export of the data from the elelconic rating plate with the "motor type" or "encoder type" selection can be forced after activation.

10.6.5 Assignment of the drive data using DO variables

Immediately after the "Automatic configuration" or after the "User-specified topology", the physical drives have been assigned to the data in the archive structure. The DOs, however, are not yet assigned to the technological drive and thus to the correct NC axis.

If optimized drive, motor or encoder data has been configured in a CMC package for a DO, it must be known to which physical DO this technological data must be assigned. This means that the path address of this DO must be known. The data is assigned using DO variables.

Diff

When configuring a package in production preparation, SINAMICS data of the modular modules is extracted once as master data to ASCII files from drive archives of one or more commissioned base machines with the comparison component Diff (e.g. drive data, motor data, encoder data of the various axes, line module data, etc.). Appropriate filter settings mean this data no longer contains any topology-specific component numbers when exported.

When exporting, Diff writes a section in front of the data that provides a reference to a topology-specific PS file in the drive data archive. This reference in the section is a short form of the path, e.g. [B3_S3_PS5] instead of [BUS3.DIR\SLAVE3.DIR\PS000005.TEA].
These modular data sets can be maintained and managed in freely selectable folder structures.

Expert

When configuring in Expert , the topology-specific path data can be replaced by symbolic, topology-independent path data using a DO variable, e.g. [$(Up.doVar.psPath)]. By assigning the drive-specific DO variables to a SINAMICS-DO and the associated simultaneous value assignment of the DO variables using the package during runtime, the DO variable has a defined value and thus also a defined path. The assignment of the DO variables to a SINAMICS-DO can be done manually at runtime in a dialog or in advance during the configuration of the comparison or user-specified topology. With this method, topology-independent SINAMICS parameters (master data) can be flexibly and easily assigned to each DO, regardless of where the drive-specific DO was placed in the SINAMICS group for the current machine.

As a result of the value assignment during the execution of the package, the symbolic path data is replaced with absolute paths and the topology-independent SINAMICS data that do not contain any component numbers are assigned to the drive-specific components in the current topology.

Example

Execution step	**Result**	
Export of data from a drive archive in Diff to an ASCII file with reference to the DO in the data storage.	[B3_S3_PS5] (corresponds to the path of the data storage [BUS3.DIR\SLAVE3.DIR\PS000005.TEA])	
Replacement of the reference with a symbolic path.	[$(Up.doX.psPath)]	
	[...]	Section for path details in the data storage
	$	Replacement operator
	Up	Identification of a package variable
	doX	DO variable
	psPath	Property that specifies the path in the data storage
Assignment of the DO variables to a DO in the topology.	Manual assignment at runtime on the "SINAMICS topology" dialog page or during the configuration of the topology	
Use of the path property after assignment of the doX variables to the DO at runtime.	[$(Up.doX.psPath)] becomes [B3_S15_PS10] if the DO is on the first NX and has the DO number 10 (BUS3.DIR\SLAVE15.DIR \PS000010.TEA)	

If DO variables are assigned as reference sensor modules of a second encoder, an assignment of this encoder to the associated servo drive can be performed while executing the package by entering the component numbers specified by SINAMICS during the package processing of CMC

automatically into the parameters for the sensor module and encoder module p141[1] or p142[1] of the DO.

Other properties of DO variables are used to obtain the current topology information and optionally further process this information, e.g. the assignment of the PROFIBUS.

Properties of DO variables

The following properties are available for a DO variable with the name "doVar". The DO variable name "doVar" can be freely selected, for example "doX", "X drive", etc.

Table 10-20 Properties of DO variables

Property	Description
Up.doVar != null	Supplies "true" if the DO variable was assigned to a drive component at runtime.
Up.doVar.doNr	Supplies the number of the DO (NOTICE: The renumbering is performed beforehand).
Up.doVar.slaveNr	Supplies the number of the higher-level slave object
Up.doVar.busNr	Supplies the number of the higher-level bus object
Up.doVar.psPath	Replaces in a section the path to the PS file in the archive.
Up.doVar.dpSlot	Supplies the PROFIBUS location number, starting with "1" (only for servo of the NCU and NX).

The "Assignment of the DO variables to SINAMICS components" figure shows an example how the properties of the DO variables are assigned after the assignment for the package processing. The properties of the DO variables can only be read in the scripts.

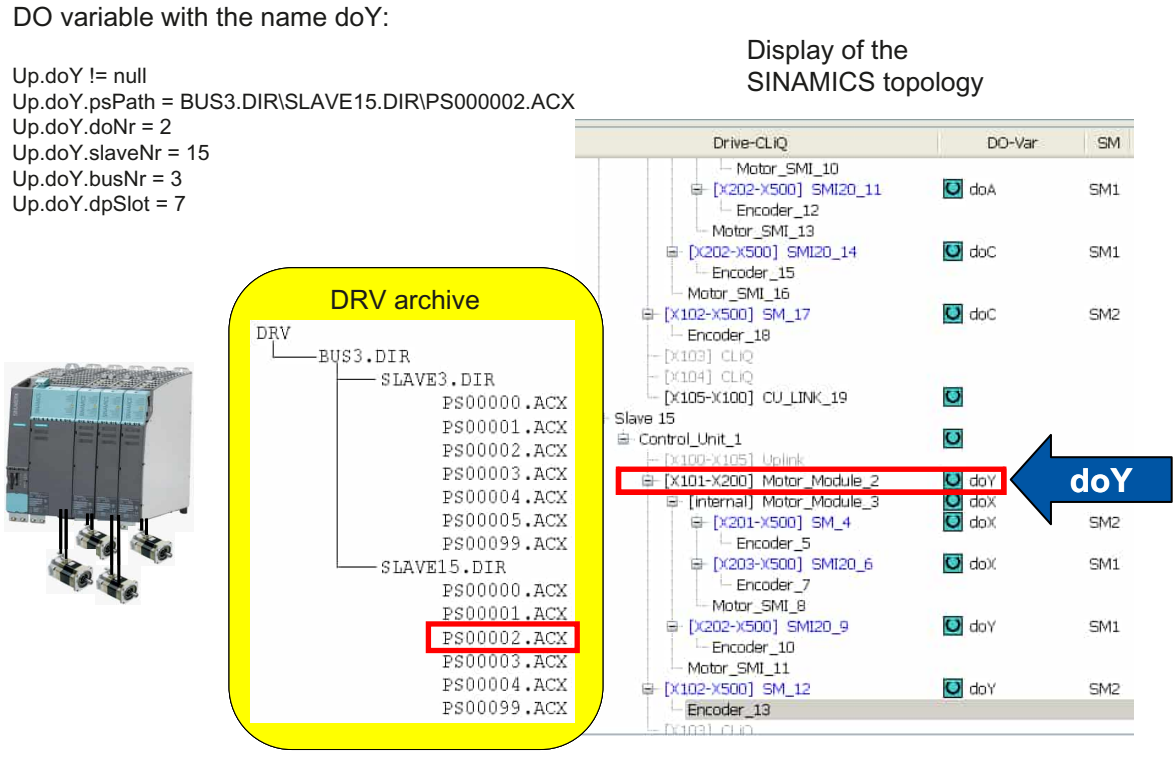


Figure 10-24 Assigning DO variables to SINAMICS components

Note

We recommend that the drive data is assigned to the drive objects via the assignment of DO variables. Although a direct assignment via a path specification is possible in short form, this limits the flexibility and can easily result in errors.

Assignment options

There are 2 ways of assigning the DO variables to SINAMICS components:

1. When processing the package, you must manually assign the DO variables to the SINAMICS components on the "SINAMICS topology" dialog page.
2. The DO variables are assigned to the SINAMICS components as a property during the configuration of a comparison topology or user-specified topology.
   The two topologies are linked into the project on the "SINAMICS topology" dialog page. Which topology is used during the runtime depends on the configuration of the DRV area on the "System configuration" dialog page.

10.6.6 The advantages of using comparison or user-specified topology

The configuration of a comparison or user-specified topology with Topo and the use of the file in the package, offer a number of additional advantages.

Advantages for the configuration of the topology with Topo

- Understandable topology creation thanks to a graphical wiring diagram
- Support with component box, shortcut menu, check run and graphic / color-coded visualization
- Printing the wiring diagram of the SINAMICS components with legend for documentation purposes
- Comparison of topologies
- Import of topologies from drive archives
- Management of DO variables

Advantages of the topology file during the package execution

- The actual wiring of the SINAMICS in the control cabinet can be checked based on the machine topology exported from SINAMICS. Warning or error messages are issued should deviations occur.
- You can check the properties of the components wired via DRIVE-CLiQ (DO type, article number, etc.). Wildcards are allowed when specifying the article number, e.g. "1FK7042 xAF7x xFxx".

- Sensor Modules not directly wired to a Motor Module, but rather to a hub or a CU, can be uniquely assigned to a DO.
- The specification of an OEM ID is used to assign the SINAMICS-CU local component number to a particular OEM global unique identifier. During package execution, the OEM component number in DO1 is written to parameter "p7859" [Component number].
- For the comparison topology, the DO names, DO numbers and component names specified by SINAMICS for the " Automatic configuration" can be replaced by user-specific data.
  DO names
  If a DO name change has been configured, then it is automatically entered in SINAMICS parameter "p199".
  DO number
  When the drive device is being configured, SINAMICS assigns the DO numbers for all DO at the "CU" and the "NX", i.e. locally for the CU, starting with number "1".
  For a machine series, it is more transparent, for example, if the X drive, irrespective of its positioning in the SINAMICS group, always has the same number. And that the DO numbers in the SINAMICS group are unique, and do not occur several times.
  The value range for the DO numbers lies between "2" and "54".
  Component names
  You can freely assign component names. The names of the SINAMICS components are displayed on the dialogs and in the screen forms and messages of SINUMERIK Operate.
- When a user-specified topology is used, the component numbers must also be specified in addition to the DO names, DO numbers and component names.
  The use of the DO names, DO numbers and component names corresponds to the above explanations for the comparison topology.
  You must also specify the component numbers for the user-specified topology. "CU" and "NX" must be assigned the "1"; it is not permissible that the numbers are assigned twice on the "CU" or "NX".
  The value range for the component numbers lies in the range between "2" and "199".
  In Topo, you can also automatically assign the component numbers via the shortcut menu.
- When configuring a user-specified topology, you can specify the activation status of the DO and components.
  The following applies:
  - 0 inactive and available
  - 1 active and available
  - 2 not available

  The status is entered in the following parameters:
  - p105 DO activation/deactivation
  - p125 Motor Module activation/deactivation
  - p145 Sensor Module activation/deactivation

DO list - comparison topology - user-specified topology

The mentioned properties and their specification are summarized in the following table:

Properties	Set	Check	DO list in the package "*.uvdx"	Comparison topology "*.ust"	User-specified topology "*.utz2"
DO variable					
DO type					
DO number			Creation in Expert/Topo		
DO name					
Wiring	SOT	SOT/VT			
Article number	SOT	SOT/VT		Creation in Topo	
Component names					
Component number					
DO activation					Creation in Topo
Component activation					

SOT Target topology

VT Comparison topology

Figure 10-25 DO variable list – comparison topology - target topology

10.6.7 Configuration of the topology with Topo

Using Create MyConfig Topo, you can configure a user-specified topology or a comparison topology. The use of Create MyConfig Topo is described in Section "Create MyConfig – Topo".

The wiring diagram for the DRIVE-CLiQ connections and some of the properties are identical for all topologies.

With switchover from comparison topology to user-specified topology, the additional properties DO activation, component activation and component number must also be specified.

When switching over to comparison topology, the data remains stored to be able to switch back to user-specified topology. This data has no effect in the comparison topology.

For the deployment as user-specified topology, other necessary data is added internally and output as binary user-specified topology file.

To deploy a user-specified topology, it is necessary to specify a CNC software version.

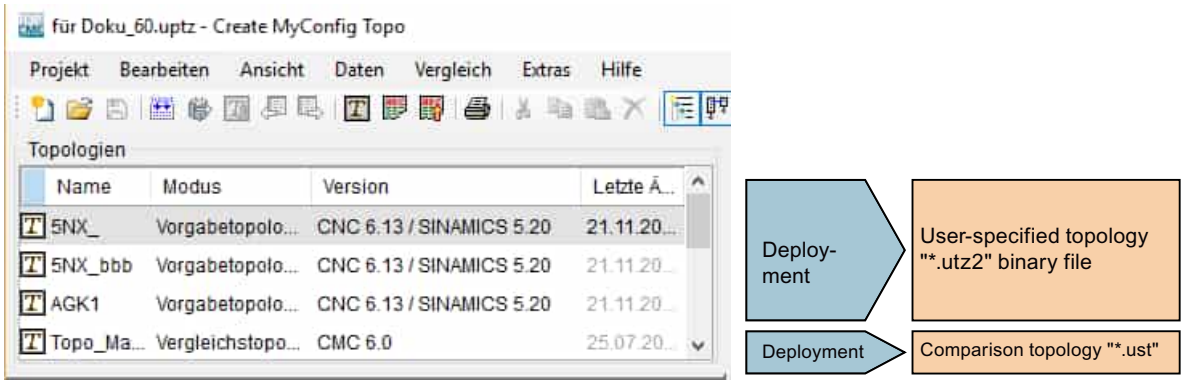


Figure 10-26 Variants of the topology generation with Create MyConfig-Topo

10.6.8 Generation of the comparison topology using external software

The structure and the contents of the comparison topology for a package is an open and documented interface, the Topo XML interface. The file format is XML.

The Topo-XML interface allows the comparison topology to also be generated using external software, which is integrated in the PLM process of the machine builder, e.g. EPLAN or ELCAD. The externally generated comparison topologies can also be imported into Topo and changed in production planning when required for checking purposes or modifications before they are integrated into the project when configuring with Expert.

On the dialog side "SINAMICS topology", during the runtime, you can also manually select a comparison configuration file, which lies in or next to the package.

Generally, you create a special topology for each machine variant, while one package can be sufficient for all machines.

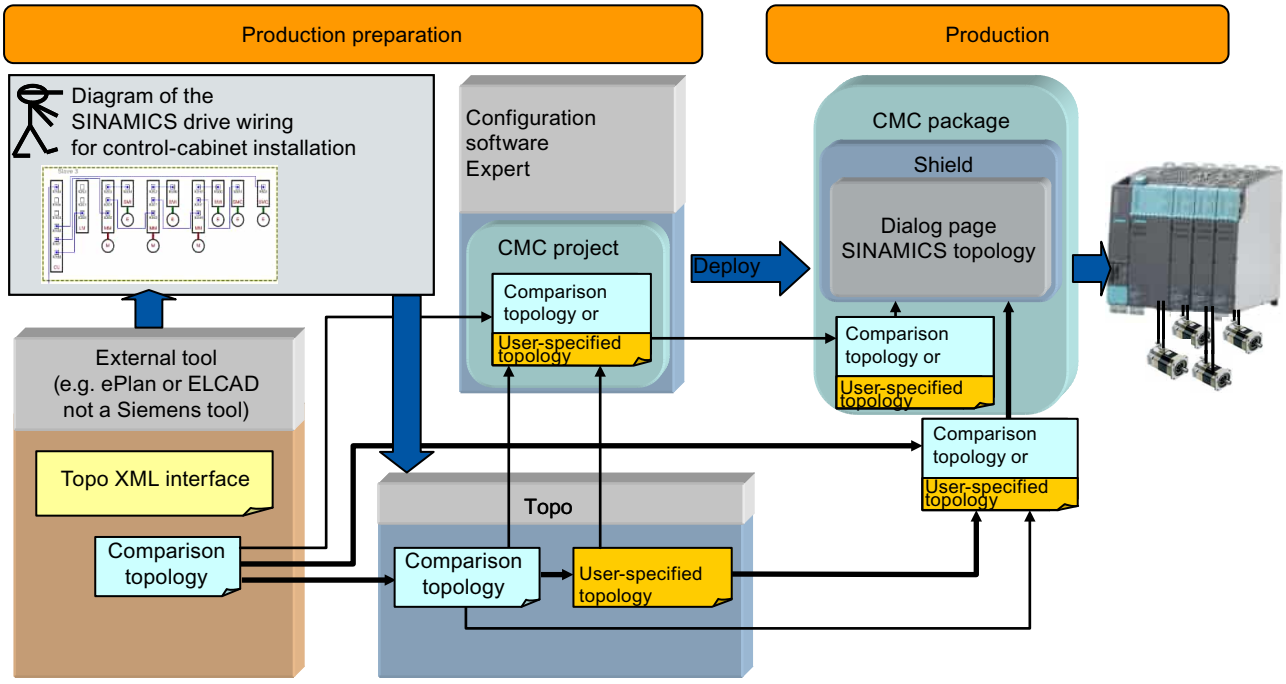


Figure 10-27 Configuration of a CMC package with comparison topology or user-specified topology

FAQs 11

11.1 Introduction

Note

Documentation structure: Operating Manual / Online Help

Section **FAQs** has the same layout and structure in the Operating Manual and online help.

11.2 Password handling for CMC V6.3 not supported

Take the following note for CMC V6.3 into account:

Note

PLC projects with activated PLC protection levels or F-passwords are not supported

PLC protection levels or F-passwords (TIA Portal: "password for modifiying safety program") are not supported for CMC 6.3 and CNC-SW6.15 and lower.

For filling using CMC V6.3 and CNC-SW6.15 and lower, do not use PLC projects with active PLC protection levels and F-passwords.

Instead, use the password-protected HMI Operate access levels "Manufacturer" or "Customer" to protect the safety configuration in the control.

If the control currently has a loaded PLC project which is protected by the passwords as described, CMC V6.3 can neither create a backup of the PLC portion nor update the current PLC configuration.

In the documentation SIMATIC Programming and Operating Manual "SIMATIC Safety - Configuring and Programming" (https://support.industry.siemens.com/cs/de/en/view/54110126) for TIA Portal V17, the section "Access protection" defines administering a PLC access level and the definition of an F-password in the "Safety Administration Editor" in order to protect a PLC safety configuration from unauthorized modification.

Within the framework of the SINUMERIK ONE STEP 7 Toolbox V17, you can also apply these measures to the integrated PLCs of the SINUMERIK ONE. Exporting a respective DSF-PLC archive from the TIA Portal is possible.

In SINUMERIK V6.15, you can also read in an access-protected DSF-PLC archive directly at the control via HMI Operate while taking the defined passwords for PLC protection levels and F-password into account.

11.3 Restoring the original status of a USB memory stick

To be able to use a USB memory stick for CMC, you must make sure that the partitions are created correctly.

A USB memory stick can be used without problems on a PC, whereas it cannot be used to run a CMC package on an NCU/PPU. This may be due to the partitions or the file system on the USB memory stick.

Prerequisite

Observe the following information about formatting a USB memory stick:

- No special program is required on a Windows PC
- Administrator rights are required

Formatting a USB memory stick

Carry out the following instructions if your USB stick does not start with a CMC package on an NCU/PPU or if similar problems arise:

```
diskpart
list disk
sel disk <n>
clean
create part primary
format quick fs=fat32 label="CMC"
```

On a Windows PC, you enter the instructions at the command prompt. On LINUX PCs, use other options such as "gparted".

Procedure

1. Open the command prompt on the Windows PC and enter `diskpart`. The utility for partitioning hard disks and removable data storage media is started in a new window. It is possible that a Windows UAC prompt appears, which you must confirm with "Yes".
2. The data storage media detected are displayed after you enter `list disk`. Here, you must find out the number of the data storage medium on the basis of the USB memory stick's memory capacity. In our example, it is data storage medium 1.
3. `Sel disk 1` selects this data storage medium. You must use the right number on the basis of your existing data storage medium.

NOTICE
Select Sel disk A data storage medium that you did not want could be destroyed if you specify a wrong number.

4. `clean` erases the data storage medium chosen via the number. All data on it is erased!

5. `create part primary` creates a primary partition on the data storage medium.
6. `format quick fs=fat32 label="CMC"` runs fast formatting with fat32. The stick is given the name "CMC".

11.4 Important folders and files

The following table contains a list of files and folders that can be helpful when using Create MyConfig (CMC).

Directory/file	Description
`%tmp%\UPShield\`	Working directory for Shield on Windows system
`tmp/sci/`	Working directory for Shield on Linux systems
`tmp/drv01/`	Directory of the USB stick on Linux systems. When using an NCU boot stick, valid after the first "Reboot from SD".
`data/`	Directory of the USB stick on Linux systems when using an NCU boot stick. Valid until the first "Reboot from SD".
`*.htm`	Logbook of the package run for diagnostic purposes through the user or SINUMERIK hotline. It is created during the package execution in the working directory of Shield and is saved at the end of the package, as configured. In addition, the logbook from Shield is always automatically saved: • When Shield runs on the NCU, in "/install" • When Shield runs on the PCU, in "D:\UP"
`SciSvcError.tar.gz`	Log file for error diagnostics through the SINUMERIK hotline. It is created next to the package.
`user/sinumerik/hmi/log/sltrc/uptrace.out`, `user/sinumerik/hmi/log/sltrc/uptrace.old`	Trace file for error diagnostics through the SINUMERIK hotline.

11.5 Simulating a project based on offline archives

Precondition

A package that has been configured for the use of an NCU or PCU, is simulated under certain constraints without the real existing NCU or PCU.

Constraints

Take the following constraints into account:

- All configurations that refer to a non-existing NCU are not processed by the package during simulation.

  Note

  The NCU is not simulated.

- If archives are available, the configurations relevant for archive data are executed.
- The "System configuration" dialog is only available when a real NCU is used, not with deactivated data area of the "NCU/PPU".
- An automatic configuration of the drives or a user-specified topology is not simulated directly. However, if you use an appropriate drive archive as initial offline archive, you can simulate the "SINAMICS topology" dialog in the "Comparison topology" version.
- Control versions of the NCU are not evaluated and not simulated.

Procedure

To simulate a package without using a real NCU/PPU or PCU, in Expert in the "Project Explorer", use the subsequently described settings.

Configuration

- In "Configuration", set the properties group "Deploy" > "Runtime system" > "Windows package (PC/IPC)".
  A package is always only simulated under Windows.
- Since the NCU/PPU is not really available, select "No" for "Use NCU/PPU" at "Data areas".
  All package parts that are relevant for the NCU are therefore ignored for the package execution.
  The NCU is not simulated.

Dialog pages

- Store the archives that serve as basis for all archive data manipulations in the "Select offline archive" dialog.
  Use a "NPDHSL" archive of your control, e.g. "Backup_NPDHSL_2019-10-25.dsf", in parts or as a whole, from the initial state before the package is executed.

Further procedure

- Deploy the package.
- Start the package.
- If the dialogs do not run automatically, edit the dialogs manually.
- The "Update.arc" result archive is next to the package after the Package execution.

11.6 First encoder of a drive on the CU or downstream of the HUB

Automatic configuration not possible

If during a drive data generation via "Automatic configuration" a first encoder (SMI or SMC) is connected downstream of a HUB or to the CU, then a complete automatic configuration of the drive by SINAMICS and Shield is not possible.

Parameter "P1982" (Pole position identification on) is set to "1" by SINAMICS instead of "0" and is not corrected automatically.

Remedy

The user can correct the parameter manually or in the step tree of a CMC package.

Set parameter "P1982=0" in the step tree in a component for all drives whose first encoder is connected downstream of the HUB or to the CU.

11.7 Parallel connection of Active Line Modules (ALM)

Precondition

The following versions are required:

- Create MyConfig V6.1
- SINUMERIK ONE V6.14
- SINAMICS V5.20

Active Line Module

As of SINAMICS 5.20 you can connect 2 Active Line Modules (ALM) to a CU or NX. The Active Line Modules are operated in parallel connection. The parameters of both ALM are saved in one drive object (DO).

The required function module is created automatically in all commissioning variants. This function is only possible with components greater than 55 kW. The deployed components must have the same power parameters.

Commissioning of this configuration is possible with:

- User-specified topology
- Automatic SINAMICS device configuration

Connect the Active Line Module to a line as shown in the following diagram.

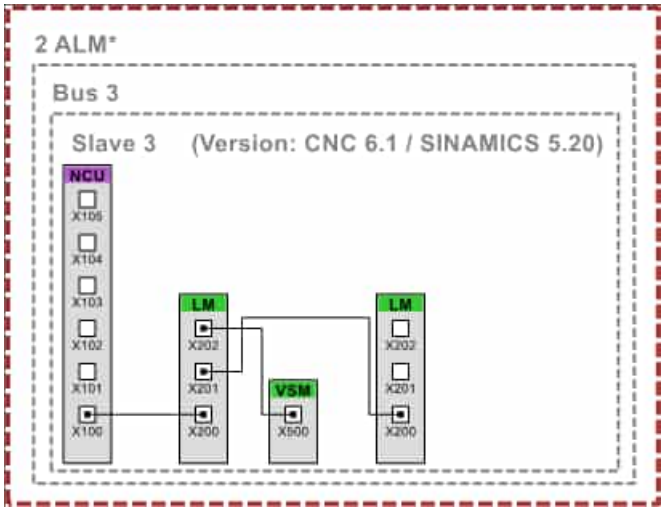


Figure 11-1 ALM configuration

You can optionally connect a VSM module as shown in the graphic.

11.8 SINAMICS Starter: No access to drive after CMC package

After the CMC package has been executed, SINAMICS Starter cannot access the drive

When configuring components that are not ordinarily available ("p105 == 0" or "p125 == 2"), problems can be incurred when SINAMICS Starter attempts to access the drive device.

Remedy

The problem exists until the component is installed for the first time. A subsequent problem involving SINAMICS Starter can be circumvented by deleting the Article No.

11.9 Certain package names result in a Windows error message

Avoiding the Windows message "This program might not have installed correctly"

Windows packages that have the designations "install", "installation" or "setup" in their names result in a confirmation prompt from Windows when starting the package on a PCU. As a rule, Windows signals that the program has not been installed correctly.

Remedy

Never use the terms "install", "installation" or "setup". in package names

For example, avoid package names such as "Install.exe", "Test_Install.exe".

List of abbreviations

A

A.1 Abbreviations

The list of abbreviations includes abbreviations used in Diff, Expert and Topo.

AAZ	Axis-drive assignment
ADA	Axis Drive Assignment
AGK	Automatic device configuration (per HMI)
ALM	Active Line Module
AML	Automation Markup Language
AP	User program
AX	Axis data
Equip. ID	Equipment identifier
CMC	Create MyConfig
CMVM	Create MyVirtual Maschine
CNC	Computerized Numerical Control
Create MyESW	Create MyEngineering & Service Workflow
CU	Control unit
DB	Data Block
DO	Drive Object
DRV	Drive Data
DRVTopo	Drive topology
FB	Function Block

FC	Function
F-PLC	Failsafe Programmable Logic Control "F" stands for "failsafe", which means that SINUMERIK Safety Integrated can be used
GUD	Global User Data
HWCN	Hardware configuration
HW Config	Hardware configuration (Step 7)
IPC	Computer that is used in the industrial environment, e.g. process visualization, robotics, industrial automation, test stands for industry or safety equipment and quality assurance
MoMo	Motor Module
NC	Numerical control
NCK	Numerical control kernel
NCU	Numerical Control Unit /Controller (example: NCU 720.2)
OB	Organization Block
PC	Personal Computer
PG	Programming device
PLC	Programmable logic controller
PPU	Panel Processing Unit: SINUMERIK 828D PPU combines, in one component, CNC, PLC, HMI, closed-loop drive control and communication functions.
RTS	RunTimeSystem (runtime system)
SDB	System data block
SIC	Safety Info Channel
SCC	Safety Control Channel
SM	Sensor Module
SMC	Sensor Module Cabinet

TCU	Thin Client Unit
UAC	User Account Control - user account control running under Windows
VT	Comparison topology
WMA	Workflow Main Application

Table A-1 File extensions for the file formats used in CMC

*.arc	NC, PLC, SDB or DRV archive
*.ard	Data class archive
*.dsf	SINUMERIK archive
*.mcl	Machine concept list
*.sdx	Siemens Data Exchange (from EPLAN Add-In)
*.tgz	CNC software
*.ucz	Expert component
*.upact	Action file Actions of the type "Copy", "INI", "Create with preprocessor", "Delete", "Execute"
*.upproj	Expert project
*.upscr	Script file: External scripts can be used in change orders and in Diff.
*.upstep	Step project in Expert
*.uptz	Topology project
*.ust	Comparison topology
*.usz	Executable Linux package

*.utz2	User-specified topology
*.uvdx	DO list

Index

"

"EPLAN Plugin for CMC" license, 421
"Siemens Industry Online Support" app, 15

#

#include, 30

A

Access to target system
  Access to target system, 263
  Configuration, 259
  System variables, 261
AccessData, 385
Action
  Delete, 373
  Execute, 375
Action of the type Copy
  Linking the directory structure, 362
Action scripts, 356
Actions
  Overview, 351
Activate step
  Example, 395
  Up.$Step[id].Activated, 394
Advantages
  Configuration of the topology with Topo, 468
  Topology file during the package execution, 468
Archive
  Step tree, 313
Archive data transfer
  Package execution, 316, 317
Archive types
  Commissioning archive, 174
  Save, 174
  SDB archive, 175
  USER archive, 175
ArcSelection, 385
Arithmetic operators, 59
Article No., 434
Assignment operators, 58
Associated components, 417
Automatic configuration
  Basic procedure, 464
  Comparison user-specified topology, 462
  First encoder, 477

B

Backup initial state
  Configuration, 273
  System variables, 275
Backup result state
  Configuration, 318
  Package execution, 321
  System variables, 319
Basic dialog settings, 220
Basic step settings, 332
Basic step settings / Step project toolbar, 332

C

Call
  Online help, 222
Call examples
  FileExist, 68
CHANDATA(1), 136, 210
Change INI
  Action, 364
Change task
  Link, 366, 370
  New, 366, 370
Character classes, 207
Check
  Topology, 420
Check run
  Project, 420
ClrBit()
  Function, 105
CMC scripts
  System variables, 34
  Up.$Env.BatchMode, 34
  Up.$Script.ArcFile, 34
  Up.$Script.EditMode, 34
  Up.$Script.Result, 34
  Up.$Script.ScriptFile, 34
CNC software
  Configuration, 279
  System variables, 280
Collapse step
  Example, 397
  Up.$Step[id].Collapsed, 397

Column layout
Mirrored, 169
Same order, 168
Command line call
Check run, 408
Diff, 200
Expert, 347, 404, 408
Notes, 202, 347, 404, 408, 412, 456
Package deployment by Expert, 404
Package execution, 411
Shield, 411
Step project export, 347
Topo, 455
Command line mode
Step project, 347
Comments, 49, 366
Commissioning engineer / operator
Access to target system, 263
Archive data transfer, 317
Backup initial state, 276
Backup result state, 321
CNC software, 283
Notes on the package, 257
Package configuration, 255
PCU access, 278
Select offline archive, 272
Step tree, 314
System configuration, 298
Version Display, 311
Comp. no.
Component number, 435
Comp. status
Component status, 435
Comparison - main menu
Topo, 424
Comparison objects, 154
Comparison operators, 60
Component
Name, 435
Replace, 431
Component names, 418
Component number
Assign, 431
Comp. no., 435
Component status, 431
Comp. status, 435
Component type, 418
Components
Differences, 452
Properties, 434, 452
Configuration
Access to target system, 259
Backup initial state, 273
Backup result state, 318
CNC software, 279
Data task, 140
Default setting, 190
File comparison, 189
File identification, 196
Finish dialog, 322
Folder comparison, 169
NC data comparison, 127
Notes on the package, 256
Package configuration, 253
PCU setup, 277
Select offline archive, 266
SINAMICS topology, 299
Step tree, 312
System configuration, 285
System variables, 237
Up.$Pack, 383
Version Display, 309
Configuration - folder comparison
Comment tab,
Configuration - folder comparison
Comparison tab,
Configuration - data task
Description tab, 143
Export tab, 142
Import tab, 143
Task type tab, 141
Template tab, 141
Configuration - file comparison
Alignment tab, 193
Comment tab, 194
Comparison tab, 190
Filter tab, 191
Configuration - file identification
Comment tab, 197
Settings tab, 196
Configuration - folder comparison
Filter tab, 172
Configuration - SINUMERIK archive comparison
Comment tab, 133
Display tab, 129
Expert list tab, 129
Filter tab, 131
Value comparison tab, 133
Configuration Expert, 462
Configuration of the topology with Topo
Advantages, 468
Configuration selection, 222
Configure comparison
Manage, 169, 189

Constructs, 208
Conversion
Import, 146
Overview, 147
Conversion tasks, 140
Copy
Clipboard, 173
NC axis machine data, 136
SINAMICS data, 135
Create with preprocessor
Action, 368

D

Data - main menu
Topo, 424
Data class archive, 148
Data matrix code, 15
Data structure tree
Main nodes, 129
Data task
Configuration, 140
Data task wizard, 144
Data tasks
Application examples, 140
Supplementary conditions, 140
DateTime
Function, 99
Default setting, 129, 171
DeployTarget, 383
Description
DO, 436
Dialog
Archive data transfer, 317
Backup initial state, 276
Backup result state, 321
CNC software, 283
Finish, 325
Step tree (package execution), 314
System configuration, 298
Version Display (package execution), 311
Dialog pages
Dependency on data areas, 241
Executing scripts, 245
Dialog script
Create, 243
Link, 244
Paste, 251
Rename, 243
Dialog scripts
Execution, 245
Dialogs
Overview, 240
Diff
Command line call, 200
Data class archive, 148
Example, 206
Introduction, 107
Parameter, 203
Return values, 206
Switch, 203
Diff main menu
Comparison, 112, 157, 178
Edit, 110, 111, 155, 156, 176, 177
File, 109, 110, 155, 176
Help, 116, 161, 183
Tools, 113, 114, 115, 158, 159, 160, 179, 180, 181, 182
View, 111, 156, 177
Window, 116, 161, 182
Differences
Components, 452
Directories, 154
Display/import topology
Project, 421
DO
Name, 418
Number, 418
Properties, 425
Variable, 418, 425, 436
DO list
DOList Editor, 439
Manage, 439
DO name, 436
DO no., 436
DO status, 431, 436
DO type, 436, 441
DO variable
Properties, 440
DO variables
Assignment options, 468
Properties, 467
DO/DO properties
Generate, 431
DOList Editor, 436
File, 437
Help, 438
Tools, 437
DOVar
Function, 100
DOVar("<doname>"), 100
DoVar(<axis>), 100

Downlink port
Rewire, 430
DQ topology export, 445
Drive data, 54
DriveTopology, 388
DrvSource, 392

E

Edit
Filter entries, 172
Edit - main menu
Topo, 423
Encoder
Number, 436
Enumeration
DeployTarget, 383
DrvSource, 392
HmiSource, 393
Mode, 391
NcSource, 391
PlcSource, 392
ProcessMode, 390
SysSource, 393
Target, 390
EPLAN, 445
Predefinition of axis assignment, 451
EPLAN Add-In, 421
Error
Procedure, 79
Escape sequence, 207
Event
Meaning, 245
OnEnd, 245
OnInit, 245
OnNext, 245
Events, 242
Example, 67
Comment, 49
DateTime, 99
Diff, 206
Expert, 407, 411
FileDelete, 67
FileExist, 68
Input, 81
InputChoice, 82
InputEnum, 84
InputInt, 86
InputReal, 87
InputText, 88
InputUInt, 89
Package execution, 413
QueryIni, 73
QueryXml, 75
StringLen, 62
StringMatch, 63
StringPos, 64
StringReplace, 65
StringSubStr, 65
Topo, 460
Examples
Programming options, 90
Execute
NCU, 377
Runtime system, 377
Executing scripts
Dialog pages, 245
Expert
Action scripts, 356
Command line mode, 404, 407
Example, 407, 411
Overview of user interface, 214
Parameter, 348, 405, 409
Return values, 406
Script Editor, 29
Step project, 347
Switch, 348, 405, 409
System variables, 380
ExpertCmd
Return values, 409
Export tasks, 140
Expression
Asterisk, 172
Question marks, 172
ExtCall
Procedure, 97

F

FAQ
Package names, 478
File comparison
Automatic file identification, 195
Comparison main menu, 178
Configuration, 189
Edit main menu, 176, 177
File main menu, 176
Help main menu, 183
Overview, 175
Overview window, 187
Toolbar, 184
Tools main menu, 179, 180, 181, 182
View main menu, 177
Window main menu, 182

File identification
Configuration, 196
File operations, 66
FileCopy, 67
Function, 66
FileDelete
Procedure, 67
FileExist
Function, 67
FileRead
Function, 69
FileWrite
Procedure, 70
Filter
SAFE, 132
Finish
Configuration, 322
Package execution, 325
System variables, 323
Folder comparison
Archive types, 174
Comparison main menu, 157
Configuration, 169
Edit main menu, 155, 156
File main menu, 155
Help main menu, 161
Overview, 154
Overview window, 165
Toolbar, 162
Tools main menu, 158, 159, 160
View main menu, 156
Window main menu, 161
Function
ClrBit(), 105
DateTime, 99
DOVar, 100
FileCopy, 66
FileExist, 67
FileRead, 69
Input, 80
InputChoice, 81
InputEnum, 84
InputInt, 85
InputReal, 86
InputText, 88
InputUInt, 89
IsBit(), 104
MathRound, 101
QueryIni, 72
QueryXml, 74
SetBit(), 105
StringLen, 62
StringMatch, 63
StringPos, 63
StringReplace, 64
StringSubStr, 65
Version, 102
Functions
Notes, 48
Usage, 47

G

General Data Protection Regulation, 16

H

Help - main menu
Topo, 425
HmiSource, 393

I

If control structure, 55
If-Else control structure, 55
If-ElseIf-Else control structure, 55
If-ElsIf control structure, 55
Import tasks, 140
Input
Function, 80
Input function, 80
InputChoice
Function, 81
InputEnum
Function, 84
InputInt
Function, 85
InputReal
Function, 86
InputText
Function, 88
InputUInt
Function, 89
Insert script
Arithmetic operators, 59
Assignment operators, 58
Comparison operators, 60
Logic operators, 60
Special operators, 61
IPC
Network connection, 403
IsBit()
Function, 104

L

Library
Utilities, 40, 104
Link new file, 374
Linux, 378
Literal characters, 207
Load standard data, 402
Log
Procedure, 101
Logbook
"*.htm", 324
Storage location, 324
Logging
Procedure, 101
Logic operators, 60

M

Main nodes
Data structure tree, 129
Manage axis-drive assignment
Topo menu, 424
Manage DO list
Topo menu, 424
MathRound
Function, 101
Memory reset, 402
Meta characters, 207
Alternation, 208
any character, 207
ASCII character, 208
Decimal number, 208
Hyphen, 208
Letter, 208
Number, 208
Spaces, 208
Square bracket, 207
Tab stop character, 208
Tabulator, 208
Underscore, 208
Unicode character, 208
Mode, 391
Module
Convert, 430
Replace, 430
Rewire, 430
Select, 432
Wiring, 430
Module catalog, 433
Msg
Procedure, 78
mySupport documentation, 13

N

Name
Component, 435
NC data
Diff, 107
NC data comparison
Configuration, 127
NcSource, 391
NCU
Network connection, 403
NCU/PPU
Package execution, 402
NcuOrigin, 386
NcuResult, 389
NcuSetup, 387
Network connection
IPC, 403
NCU, 403
New file, 374
Notes on the package
Configuration, 256
Package execution, 257
System variables, 257

O

OEM ID, 435
OnEnd, 245
OnInit, 245
Online help, 222
OnNext, 245
OpenSSL, 16
Operator
Arithmetic operators, 59
Assignment operators, 58
Comparison operators, 60
Logic operators, 60
Operators, 367

P

Package
Simulation, 475
Package configuration
Configuration, 253

Package execution, 255
System variables, 254
Package execution
Archive data transfer, 316, 317
Backup initial state, 276
Backup result state, 321
Buttons, 255, 258
CNC software, 283
Example, 413
Finish dialog, 325
NCU/PPU, 402
Notes on the package, 257
Package configuration, 255
Parameter, 412
PCU setup, 278
Return values, 413
Select offline archive, 272
SINAMICS topology, 304
Step tree, 314
System configuration, 298
Version Display, 311
Package names
FAQ, 478
Package variables, 44
Up variables, 44
PackageConfig, 384
PackageEnd, 389
PackageNotes, 384
Parameter
Diff, 203
Expert, 348, 405, 409
Package execution, 412
Topo, 458
Paste
Dialog script, 251
Script, 251
PCU setup, 386
Configuration, 277
Package execution, 278
PlcSource, 392
Procedure
Error, 79
ExtCall, 97
FileDelete, 67
FileWrite, 70
Load standard data, 402
Log, 101
Logging, 101
Memory reset, 402
Msg, 78
Redo, 98
Return, 97
Skip, 98
TraceToFile, 76
Warning, 79
Procedures
Notes, 48
Special feature, 47
ProcessMode, 390
Product support, 14
Program elements
Color coding, 33
Programming examples, 90
Project
Check run, 420
Display/import topology, 421
Project - main menu
Topo, 420, 421, 422
Project main menu
EPLAN, 421

Q

Quantifier, 209
Query step
Example, 394
QueryIni
Function, 72
QueryXml
Function, 74

R

Redo
Procedure, 98
Regular expressions
Examples, 209
Replacement operator
Special operators, 61
Return
Procedure, 97
Return values
Diff, 206
Expert, 406
ExpertCmd, 409
Package execution, 413
Topo, 460
Rewire, 430
Downlink port, 430
Uplink port, 430

S

Script
#include, 30
Paste, 251
Script call
Switch, 204
Script Editor, 366
Expert, 29
Script language
String concatenation, 62
Script libraries, 236
Scripts
Actions, 32
Select offline archive
Configuration, 266
System variables, 268
Selectoffline archive
Package execution, 272
Send feedback, 13
SetBit()
Function, 105
Siemens DQ-Topo Export, 421
Siemens Industry Online Support
App, 15
SINAMICS data
Copy, 135
SINAMICS topology
Comparison topology, 304
Configuration, 299
Manual DO variable assignment, 307
Package execution, 304
System variables, 301
User-specified topology, 306
SINUMERIK, 11
SINUMERIK archive comparison
Comparison main menu, 112
Configuration, 127
Copying/exporting data, 134
Data task wizard, 144
Edit main menu, 110, 111
File main menu, 109, 110
Functionality, 108
Help main menu, 116
Toolbar, 117
Tools main menu, 113, 114, 115
View main menu, 111
Window, 123
Window main menu, 116
SINUMERIK archives, 173
DSF format, 154
SINUMERIK ONE, 12
SINUMERIK ONE, 107
Skip
Procedure, 98
SOT, 417
Special operators
Replacement operator, 61
Standard comparison software, 107
Standard scope, 12
Standard script libraries, 40, 104
Step
Collapsed, 335
Configuring, 333
Copy, 330
Execute, 334
ID, 334
Information text, 333
Locked, 334
Name, 333
Paste, 330
Path, 334
Single selection substeps, 335
Step activation locked
Example, 396
Up.$Step[id].Locked, 396
Step configuration, 334
Step project, 332
Command line mode, 347
Step tree
Archive, 313
Configuration, 312
Package execution, 314
System variables, 313
Steps
Toolbar, 328
Steps
Up.$Step, 393
StepSelection, 389
Storage location
Logbook, 324
String concatenation, 62
StringLen
Function, 62
StringMatch
Function, 63
StringPos
Function, 63
StringReplace
Function, 64
StringSubStr
Function, 65

Switch
Diff, 203
Expert, 348, 405, 409
Script call, 204
Topo, 458
SysSource, 393
System configuration, 461
Automatic configuration, 461
Configuration, 285
Package execution, 298
System variables, 287
User-specified topology, 461
System variables
"PCU setup" dialog - PCU setup, 386
Access to target system, 261
AccessData, 385
ArcSelection, 385
Backup initial state, 275
Backup initial state - NcuOrigin, 386
Backup result state, 319
Backup result state - NcuResult, 389
CMC scripts, 34
CNC software, 280
CNC software - NcuSetup, 387
Configuration, 237, 383
DeployTarget, 383
Display for the dialog, 221
Display of steps, 333
DriveTopology, 388
DrvSource, 392
Enumeration runtime environments - Runtimes, 399
Expert, 380
Finish, 323
for dialog Finish - PackageEnd, 389
for the Package configuration dialog - PackageConfig, 384
HmiSource, 393
Mode, 391
NcSource, 391
NcuOrigin, 386
NcuResult, 389
NcuSetup, 387
Notes on the package, 257
Notes on the package - PackageNotes, 384
Package configuration, 254
PackageConfig, 384
PackageEnd, 389
PackageNotes, 384
PCU setup, 386
PlcSource, 392
ProcessMode, 390
Select offline archive, 268
Select offline archive - ArcSelection, 385
Selection of target system - AccessData, 385
SINAMICS topology, 301
SINAMICS topology - DriveTopology, 388
Step tree, 313
Step tree - StepSelection, 389
StepSelection, 389
SysSource, 393
System configuration, 287
System configuration - SystemConfig, 387
SystemConfig, 387
Target, 390
Up.$Env, 398
Up.$Env.BatchMode, 35
Up.$Script.ArcFile, 35
Up.$Script.EditMode, 38
Up.$Script.Result, 36
Up.$Script.ScriptFile, 35
Up.$Step, 393
Version Display, 310
Version Display - VersionView, 388
VersionView, 388
SystemConfig, 387

T

TAR archives, 154, 173
Target, 390
Technical support, 15
TGZ archives, 154, 173
Toolbar
Basic dialog settings, 220
File comparison, 184
Folder comparison, 162
Project explorer, 217
Project processing, 222
SINUMERIK archive comparison, 117
Steps, 328
Toolbars
Topo menu, 423
Tools - main menu
Topo, 425
Topo
Command line call, 455
Comparison main menu, 424
Data main menu, 424
Edit main menu, 423
Example, 460
Help main menu, 425
Parameter, 458
Project main menu, 420, 421, 422

Return values, 460
Switch, 458
Tools main menu, 425
View main menu, 423, 424
Topology
Check, 420
Topology comparison, 426
Topology file during the package execution
Advantages, 468
Topology graph, 426
Topo menu, 423
Topology tree, 426
Topo menu, 423
TraceToFile
Procedure, 76
Training, 15

U

Up variables
Package variables, 44
Up.$Dialog.AccessData, 385
Up.$Dialog.ArcSelection, 385
Up.$Dialog.DriveTopology, 388
Up.$Dialog.NcuOrigin, 386
Up.$Dialog.NcuResult, 389
Up.$Dialog.NcuSetup, 387
Up.$Dialog.PackageConfig, 384
Up.$Dialog.PackageEnd, 389
Up.$Dialog.PackageNotes, 384
Up.$Dialog.PcuSetup, 386
Up.$Dialog.StepSelection, 389
Up.$Dialog.SystemConfig, 387
Up.$Dialog.VersionView, 388
Up.$Env.BatchMode
System variables, 34, 35
Up.$Pack
Configuration, 383
Up.$Pack.ARC, 382
Up.$Pack.DeployDir, 382
Up.$Pack.DeployName, 382
Up.$Pack.DeployTarget, 382
Up.$Pack.NCU, 382
Up.$Pack.PCU, 382
Up.$Pack.ProdVersion, 382
Up.$Script.ArcFile
System variables, 34, 35
Up.$Script.EditMode
System variables, 34, 38
Up.$Script.Result
System variables, 34, 36
Up.$Script.ScriptFile
System variables, 34, 35
Up.$Step, 393
Up.$Step[id].Activated, 394
Up.$Step[id].Collapsed, 397
Up.$Step[id].Locked, 396
Uplink port
Rewire, 430
User-specified topology, 418
Basic procedure, 465
Comparison automatic configuration, 462
User-specified topology (*.utz2), 302
UST, 456
Utilities
Library, 40, 104
UTZ2, 456
UVDX, 456

V

Value assignment, 134
Value table, 136
Version
Function, 102
Version Display
Configuration, 309
Package execution, 311
System variables, 310
VersionView, 388
View - main menu
Topo, 423, 424

W

Warning
Procedure, 79
Websites of third-party companies, 12
Window
SINUMERIK archive comparison, 123
Windows, 378

Z

ZIP archives, 154, 173