SIEMENS

SINUMERIK

SINUMERIK 840D sl SINUMERIK Integrate Create MyConfig - Diff, Expert, Topo

Operating Manual

Valid for:
CMC software V4.7 SP1

06/2015
A5E36537479B

Legal information

Warning notice system

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

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

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

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

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

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

Qualified Personnel

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

Proper use of Siemens products

Note the following:

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

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E36537479B
Ⓟ 08/2015 Subject to change

Preface

Introduction

Target group

This manual is intended for machine manufacturers and service technicians.

Benefits

The Create MyConfig engineering software is used by the machine manufacturer to create and run a project for automated commissioning/production of machines with SINUMERIK 840D sl controllers. Even upgrades of those controllers at the end customer can be configured and executed automatically.

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

In contrast to the conventional procedure, individual operations on the machine can be performed faster and with greater ease and reliability.

- The production time for machines with the 840D sl is significantly reduced.
- The upgrade time for end customers is also reduced.
- A high level of reproducibility is achieved through automatic data loading.
- The quality in production and service is improved significantly.
- Lower demands are placed on the expert knowledge of the commissioning engineer.
- A modular data management for modular machines is supported at the machine manufacturer.
- The modular machines are adapted to the flexible SINAMICS topologies.
- The commissioning process is documented in a log file.

The Create MyConfig engineering software comprises the four components Diff, Expert, Topo and Shield.

- Diff is a comparison software and supports you in the provision of modular data.
- Expert is a configuration software and supports you when creating a project. Deployment of a project produces an executable package.
- Topo is a topology software and supports you with displaying and/or creating a comparison and/or target topology for commissioning of the SINAMICS drive system.
  With Topo, the axis-drive assignment can also be displayed or configured.
- Shield is the upgrade software and cannot be installed on its own. Deploying the project with Expert creates an executable package containing the software component Shield, a script, and data. Shield executes the configured instructions when the package is started.

Training

For information about the range of training courses, refer to:

- SITRAIN (www.siemens.com/sitrain) - training courses from Siemens for automation products, systems and solutions

FAQs

You can find Frequently Asked Questions in the Service&Support pages under Product Support (www.siemens.com/automation/service&support).

SINUMERIK

You can find information on SINUMERIK under the following link: (http://www.siemens.com/sinumerik)

Technical Support

Country-specific telephone numbers for technical support are provided in the Internet under "Contact" (www.siemens.com/automation/service&support).

Table of contents

Safety information 1

1.1 Fundamental safety instructions

1.1.1 General safety instructions

⚠ WARNING

Risk of death if the safety instructions and remaining risks are not carefully observed

If the safety instructions and residual risks are not observed in the associated hardware documentation, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

⚠ WARNING

Danger to life or malfunctions of the machine as a result of incorrect or changed parameterization

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

- Protect the parameterization (parameter assignments) against unauthorized access.
- Respond to possible malfunctions by applying suitable measures (e.g. EMERGENCY STOP or EMERGENCY OFF).

1.1.2 Industrial security

Note

Industrial security

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

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

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

⚠ **WARNING**

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

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

- Keep the software up to date.
  You will find relevant information and newsletters at this address (http://support.automation.siemens.com).
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
  You will find further information at this address (http://www.siemens.com/industrialsecurity).
- Make sure that you include all installed products into the holistic industrial security concept.

Installation 2

2.1 Overview

You have two options for installing the Create MyConfig engineering software and the NcuShareService application on your PC:

- Installation using the installation wizard
  Installing the Create MyConfig setup as usual by running the setup file and by manually working through the setup dialogs.
- Installation via command line
  From version 4.6, it can also be automatically installed without any user interaction.
  An "OPFILE.txt" control file can be transferred as argument when starting the setup.
  The setup process can be preconfigured in this control file. This means that it is no longer necessary to manually work through the setup dialogs. The following chapter describes the option for installation via the command line.

2.2 Installation via command line

2.2.1 Structure of the "OPFILE.txt" control file

The file contains the section data as first entry [SETUPREQUESTS].
A parameter value is then allocated per line.
Comment lines can be started with the "#” character:

Parameter description

Parameters and their possible values are listed in the following table:

Parameter	Description	Example
SetupMode	The "Batch" value must be allocated to the parameter in order to prevent setup dialogs being displayed.	SetupMode=Batch
DestinationDrive (optional)	The target drive where the software should be installed is specified in this parameter. The system drive is used if a value is not entered. This parameter is ignored if CMC components are already installed.	DestinationDrive=C

Parameter	Description	Example
DestinationPath (optional)	A path without a specific drive where the software should be installed can be specified in the parameter. A default installation path is used if a value is not entered. Under a 32-bit Windows operating system, for example, the path is "\Program Files\Siemens\Create MyConfig 4.7." This parameter is ignored if CMC components are already installed.	DestinationPath ="\myTools \Create MyConfig 4.7"
Features (optional)	The components to be installed, separated by commas, are listed in this parameter. Possible values include Expert, Topo, Diff and Samples. The components correspond to the selection in the setup dialog when installing manually. All of the components are selected if a value is not entered.	• Install all components: Features=Expert,Topo,Diff, Samples • Only install "Diff" components: Features=Diff
DeleteSettings (optional)	With this parameter, it is possible to force deletion of settings from a previous installation. If the optional parameter is omitted, the prior settings from a previous installation will be retained.	DeleteSettings=1

Example of the "OPFILE.txt" control file

```
[SETUPREQUESTS]
# Do not display setup dialogs
SetupMode=Batch
# Target drive for installation
DestinationDrive=C
# Target path for installation
DestinationPath="\Program Files\Siemens\Create MyConfig 4.7"
# Component selection for installation
Features=Expert,Topo,Diff,Samples
#
#
# Delete settings from previous installation
# (default: DeleteSettings=0)
# DeleteSettings=1
#
# Installing the NcuShareService
# (default: NcuShareService=1)
# NcuShareService=1
#
#
# Selection of the share folder of the NcuShareService
# NcuShareServiceShareFolder=c:\Siemens\NcuShare
```

Note

In addition to the setup file, the commented template "OPFILE.txt" is also stored, which can be adapted and used accordingly.

Example of the "OPFILE.txt" control file for installation with standard settings

```
[SETUPREQUESTS]
SetupMode=Batch
```

2.2.2 Command line call

Syntax for the command line call

```
setup.exe /W /L<Language ID> /V"commandline=\”\”\”-OF:<absolute path
\>OPFILE.txt\”\”\””
```

Options and arguments

Options	Arguments and description
L	<Language ID> The option L (language) indicates the language ID for displaying the language in the dialogs and messages. The language ID does not influence the installed program language. The following arguments are admissible: • 1031 for German • 1033 for English
W	The option does not include any arguments. Setup.exe waits until the installation has been completed, and only then is it closed.
V	<Msiexec (command line options)> Parameters can be specified as arguments to call up the Windows Installer. The argument must be specified in double inverted commas. A description of possible parameters can be found under the topic "Msiexec (command line options)" on theInternet (http://technet.microsoft.com/en-en/library/cc759262(v=ws.10).aspx).

Note

A list of all of the options and additional information can be viewed at the following link:

Command line parameter for setup.exe (https://flexeracommunity.force.com/customer/articles/en_US/INFO/Setup-exe-Command-Line-Parameters)

Example of a call for an English installation and OPFILE.txt in the setup folder

```
setup.exe /W /L1033 /V"commandline=\”\”\”-OF:OPFILE.txt\”\”\””
```

Example of a call for a German installation and OPFILE.txt with absolute path data

```
setup.exe /W /L1031 /V"commandline=\”\”\”-OF:D:\mySetupConfiguration
\OPFILE.txt\”\”\””
```

2.2.3 Modifying installed program parts

The individual installed program parts (Diff, Topo, Expert, Samples) can be easily added or removed afterwards.

There are three versions:

Version 1

1. If you have installed **Windows XP** , in the start menu, select:
   "Start > Settings > Control Panel > Software".
   If you have installed **Windows 7**, select
   "Programs and functions".
2. Then select Create MyConfig 4.7 and click on "Change" or "Change/add".
3. Select the program parts that you wish to add or remove.

Version 2

1. Start the "setup.exe" again on the CD.

You then go to the same dialogs, and you can proceed as in version 1.

Version 3

1. Use the "OPFILE.txt" control file described above to subsequently add or delete program parts.
   If, for Features, you specify another configuration, then it is precisely this one that will be installed.
   – If you have already installed "Topo", and you wish to also install "Diff", then the "Features" parameter looks like this:
     Features=Topo,Diff
   – If instead of the installed Topo, you only wish to have Diff installed, then the "Features" parameter looks like this:
     Features=Diff

These three versions can be combined. For example, this means you can first use version 1, for a subsequent change, version 3, and later, for a new change, version 2.

Create MyConfig - Diff 3

3.1 Safety instructions

3.1.1 Fundamental safety instructions

3.1.1.1 General safety instructions

⚠ WARNING

Risk of death if the safety instructions and remaining risks are not carefully observed

If the safety instructions and residual risks are not observed in the associated hardware documentation, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

⚠ WARNING

Danger to life or malfunctions of the machine as a result of incorrect or changed parameterization

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

- Protect the parameterization (parameter assignments) against unauthorized access.
- Respond to possible malfunctions by applying suitable measures (e.g. EMERGENCY STOP or EMERGENCY OFF).

3.1.1.2 Industrial security

Note

Industrial security

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

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

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

⚠ WARNING

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

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

- Keep the software up to date.
  You will find relevant information and newsletters at this address (http://support.automation.siemens.com).
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
  You will find further information at this address (http://www.siemens.com/industrialsecurity).
- Make sure that you include all installed products into the holistic industrial security concept.

3.2 Introduction

Note

Setup of documentation: Operating Manual / online help

The contents for the **Diff** section are identical in the operating manual and the online help.

Diff is software for displaying and comparing folders, SINUMERIK archives, general files and special files of the SINUMERIK 840D sl control.

With Diff, SINUMERIK archives can also be modified and stored.
Selected data can be exported from SINUMERIK archives or can be imported into SINUMERIK archives and stored.

For data class archives of the SINUMERIK 828D control, unlimited functionality is available.

Diff comprises the following three components:

- NC data comparison (Page 19) → Display and comparison of NC, drive or display data from SINUMERIK archives or files
- Folder comparison (Page 47) → Display and comparison of folders, SINUMERIK archives and archives in ZIP, TGZ and TAR format.
- File comparison (Page 60) → Display and comparison of files.

These three components can be called individually and any number of times in the application and are each displayed as a tab card. Diff can be used to support the configuration of packages with Expert, and independent of Expert, as standard comparison software.
In all three components, the opened archive, folder or files can be edited.

The data determined with Diff is provided in such a way that it can be incorporated into a project directly.
The advantage of Diff that makes it so unique is the fact that it can process SINUMERIK archives in the PC or punched tape format and special SINUMERIK files and SINUMERIK data structures.

Diff shows NC and drive data in a structured fashion in an NC data comparison.

3.3 NC data comparison

3.3.1 General information about NC data comparison.

Any number of comparison objects (SINUMERIK archives or Sub files) can be imported, displayed and compared in an NC data comparison.
The comparison is realized at the data level.

The NC data comparison supports the comparison of the following objects:

- SINUMERIK archives in the PC 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 NC data comparison extracts the NC, drive, and display data from the SINUMERIK archives and displays these 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 and the value of the last one that was imported is used.

The data 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.
Data from the objects being compared can be copied to the clipboard and then used in other applications (Expert, ...).

It is possible to remove or insert identifiers and change the values of existing identifiers in the "Selective" comparison mode.

Diff permits the user to resave an open SINUMERIK archive with modified data as SINUMERIK archive.

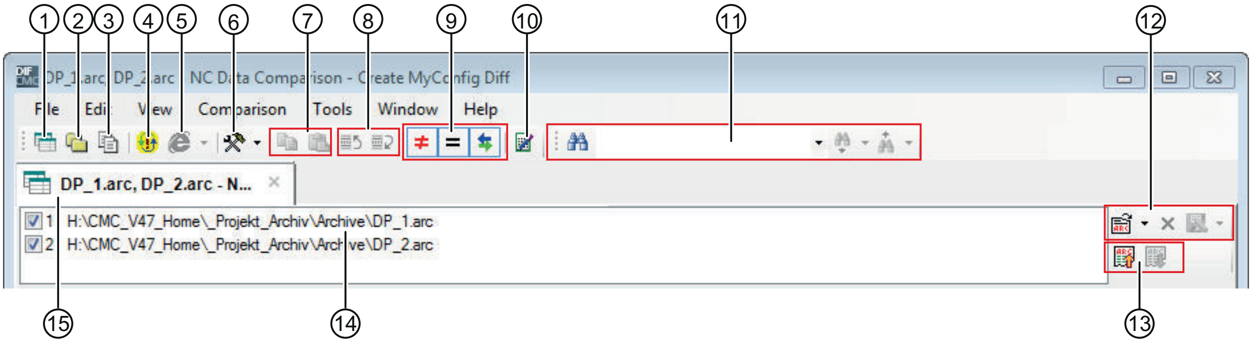
Note

In an NC data comparison, Diff always displays the data contained in the comparison object if no settings were made in the comparison configuration 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.

3.3.2 Overview of user interface

3.3.2.1 NC data comparison - toolbar


①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑩
⑪
⑫
DP_1.arc DP_2.arc NC Data Comparison - Create MyConfig Diff
File
Edit
View
Comparison
Tools
Window
Help
DP_1.arc, DP_2.arc - N...
⑮
⑭
⑬

☑ 1 H:\CMC_V47_Home_Projekt_Archiv\Archive\DP_1.arc
☑ 2 H:\CMC_V47_Home_Projekt_Archiv\Archive\DP_2.arc

① New NC data comparison

② New folder comparison

③ New file comparison

④ Update comparison

⑤ Display/save HTML/XML protocol

⑥ Comparison configuration

⑦ Copy / Insert

⑧ Go to the previous or next difference.

⑨ Different / The same / Not available in all comparison objects

⑩ Changed only

⑪ Search functions

⑫ Select archive or file for NC data comparison / Close archive or file / Save as

⑬ Create archive from control system / read into control system

⑭ List of opened comparison objects

⑮ Switch between opened comparisons via tab

Description of buttons/fields of the toolbar

The buttons of the toolbar are described in more detail below.

① New NC data comparison

Open a new NC data comparison.

② New folder comparison

Open a new folder comparison.

③ New file comparison

Open a new file comparison.

④ Update comparison

The button for updating the comparison is always shown in color if the displayed comparison result is not a current result and the comparison should be restarted using this button.

Note

Depending on the comparison situation, the comparison is automatically started by Diff in some cases.

⑤ Display/save HTML/XML protocol

Using this button, the comparison result of the node selected in the data structure (see NC data comparison - Overview window (Page 23): position numbers ① and ②) is displayed in the standard browser (Internet Explorer).

The result can be saved either as an HTML or XML report.
It can be output complete with expanded fields or as it is shown in the display.

⑥ Comparison configuration

Using this button, a comparison configuration can be changed or a different comparison configuration can be selected (see sectionConfiguration (Page 26)).

⑥ Copy / Insert

Buttons for copying and inserting (selective mode) selected values of one or more comparison objects (see section Copying/exporting of data from the NC data comparison).

⑥ Go to the previous or next difference.

The differences in direction, forward or backward, can be displayed here.

⑥ Different / The same / Not available in all comparison objects

The differing and identical parts of the comparison result or parts that are not available in all comparison objects can be hidden or displayed via these buttons.

⑥ Changed only

The display can be reduced to just show the modified values when activated using this button. This selection acts in addition to the display restrictions selected using button group ⑧ and those defined by the comparison configuration.

⑥ Search functions

This group contains:

- a button for opening a dialog box in which the search options can be defined
- an edit box for searching for character strings in selected nodes
- buttons for searching upward and downward.

Search down
Search up

⑫ Selecting comparison objects

Select archive or file for NC data comparison
▾ The last opened archives or files are listed.
✕ Close archive or file.
Button to save a modified archive.

⑥ Create online archive from control system / read into control system

Load archive from the control
Load archive into the control

⑥ List of opened comparison objects

As many objects as required can be opened for comparison.

The number at the beginning of the line is used to identify the comparison object, e.g. in the column header (see NC data comparison - overview window (Page 23): the area at position number ⑧).

Note

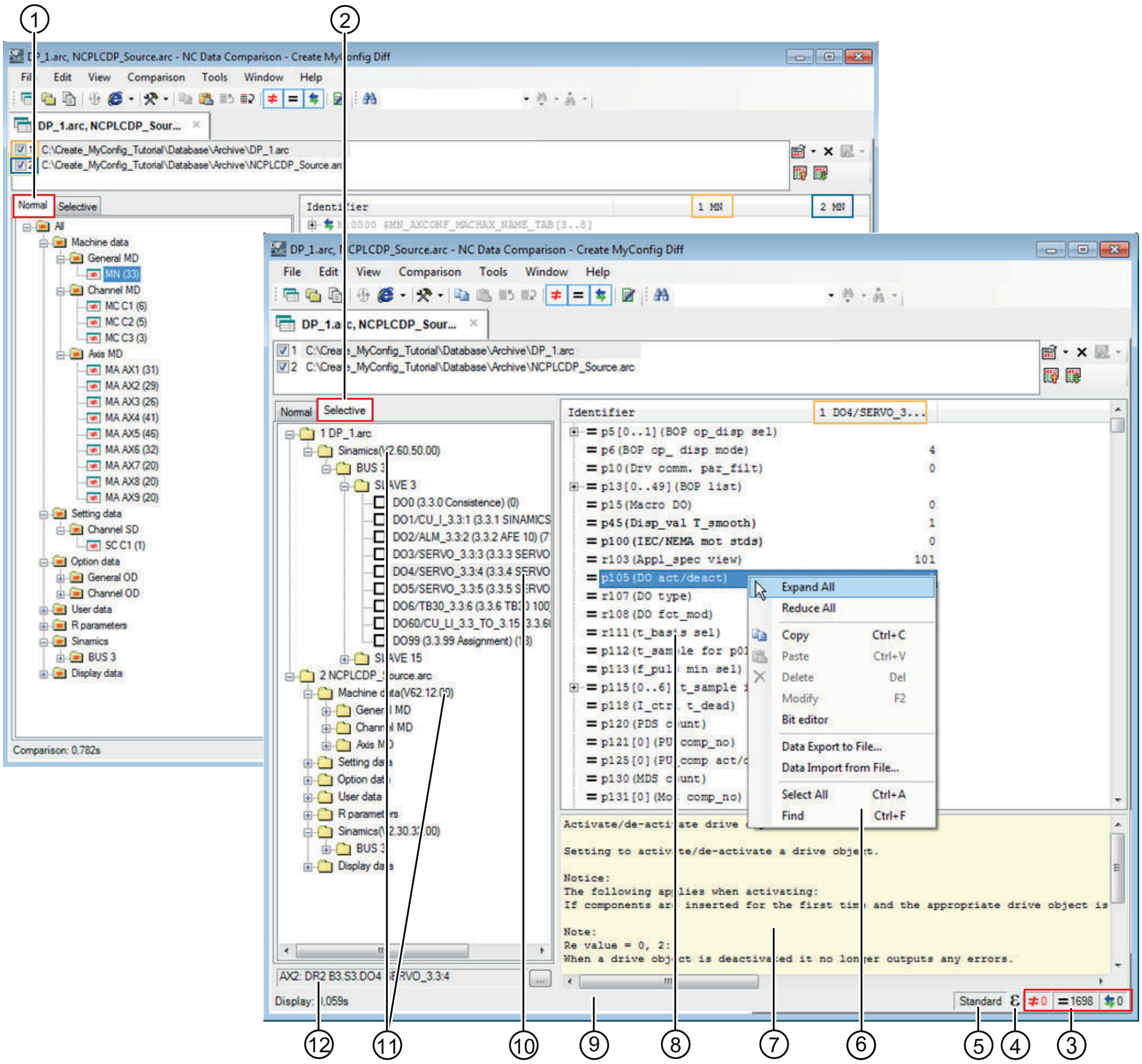
Modified archives that have not been saved are assigned an asterisk "*" in list ⑪.

⑥ 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".

3.3.2.2 NC data comparison - overview window

Below is an overview of the functions in the NC data comparison window.



☑ 1 C:\Create_MyConfig_Tutorial\Database\Archive\DP_1.arc
☑ 2 C:\Create_MyConfig_Tutorial\Database\Archive\NCPLCDP_Source.arc

```
= p5[0..1](BOP op_disp sel)
= p6(BOP op_ disp mode)          4
= p10(Drv comm. par_filt)        0
= p13[0..49](BOP list)
= p15(Macro DO)                  0
= p45(Disp_val T_smooth)         1
= p100(IEC/NEMA mot stds)        0
= r103(Appl_spec view)         101
= p105(DO act/deact)
```

Activate/de-activate drive object

Setting to activate/de-activate a drive object.

Notice:
The following applies when activating:
If components are inserted for the first time and the appropriate drive object is

Note:
Re value = 0, 2:
When a drive object is deactivated it no longer outputs any errors.

AX2: DR2 B3.S3.DO4 SERVO_3.3:4

Display: 0,059s

Comparison: 0,782s

① Normal tab

② Selective tab

③ Display result: Different / The same / Not available in all comparison objects

④ Expert list and Filter functions

⑤ Display of the comparison configuration

⑥ Context menu

⑦ A brief description of the selected parameter is displayed in the text area with yellow background.

⑧ Result from the Normal tab / Selective tab

⑨ Status bar

⑩ Short designation of data end node:

⑪ Behind the node names "Machine data" and "SINAMICS", the NCK and SINAMICS version is displayed from which the data originate.

⑫ Axis-drive assignment

Description of the areas/functionalities in the NC data comparison window

The individual functionalities of the NC data comparison are described in more detail below:

① Normal tab

In this area, the data nodes of the comparison objects are shown in a tree structure in the comparison mode "Normal" or in the comparison mode "Selective".

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 symbols.

For example, the NC data of axis 1 (AX1) are 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.

= The equal character in the area node indicates that the data of the objects being compared is identical below this area node.

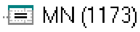
≠ The unequal character in the area node indicates that not all of the data of the objects being compared is identical below this area node.

= The equal character in the end node indicates that the data of the objects being compared is identical within this end node.

≠ The unequal character in the end node indicates that not all of the data of the objects being compared is identical within this 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:

 MN (1173)	This node contains 1173 identifiers and they are all displayed.
MN (1086/1173)	This node contains 1173 identifiers, and as a result of the Expert list or the filter, only 1086 identifiers are displayed.

① Selective tab

In this area, the data nodes of the comparison objects are shown in a tree structure in the comparison mode "Normal" or in the comparison mode "Selective".

In the "Selective" comparison mode, the data structures of all comparison objects (archives, 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.

The data, which should be compared with one another, is selected by activating the end nodes.

① Display result: Different / The same / 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).

④ Expert list and Filter functions

If the functions "Expert list" and "Filter" are active in the comparison configuration, the relevant symbols will be displayed.
With a double click, the expert list or filter in the comparison configuration is opened.

Note

Using "Expert list" and "Filter," non-relevant identifiers can be excluded from the comparison, so that these are no longer displayed.

① Display of the comparison configuration

The comparison configuration that is being used in the actual comparison is displayed in this field.

The comparison configuration is opened with a double-click.

① Result from the Normal tab / Selective tab

The data of the nodes selected in areas ① and ② 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.

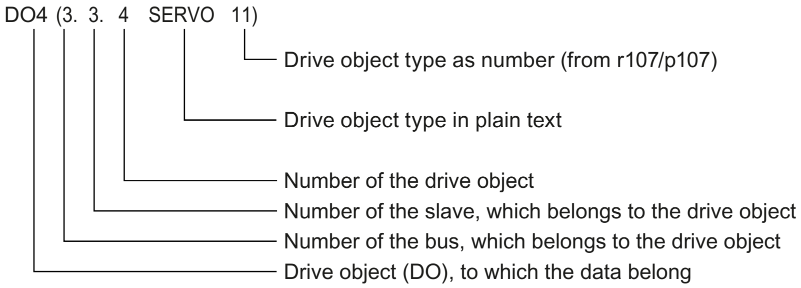
Values shown in **blue** are values that the user changed in the "Selective" mode.

The number as prefix in the column header of the values defines the comparison object (archive, file) to which these values belong.

⑩ Data end node - short designator

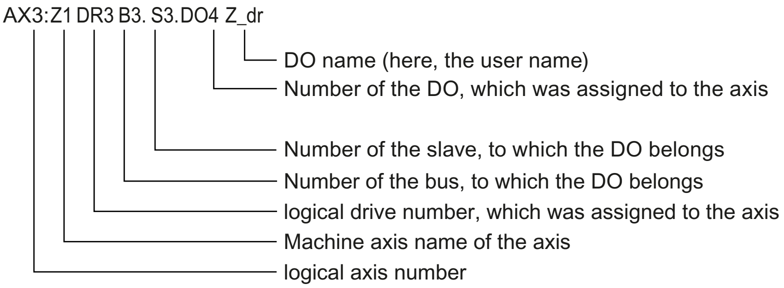
The data end nodes have a code that defines the data area from which the data originated.

For SINAMICS data, the following information is contained in the node designation.


DO4 (3. 3. 4 SERVO 11)
Drive object type as number (from r107/p107)
Drive object type in plain text
Number of the drive object
Number of the slave, which belongs to the drive object
Number of the bus, which belongs to the drive object
Drive object (DO), to which the data belong

⑫ Axis-drive assignment

Axis-drive assignment of the selected drive object or the selected axis.


AX3:Z1 DR3 B3. S3.DO4 Z_dr
DO name (here, the user name)
Number of the DO, which was assigned to the axis
Number of the slave, to which the DO belongs
Number of the bus, to which the DO belongs
logical drive number, which was assigned to the axis
Machine axis name of the axis
logical axis number

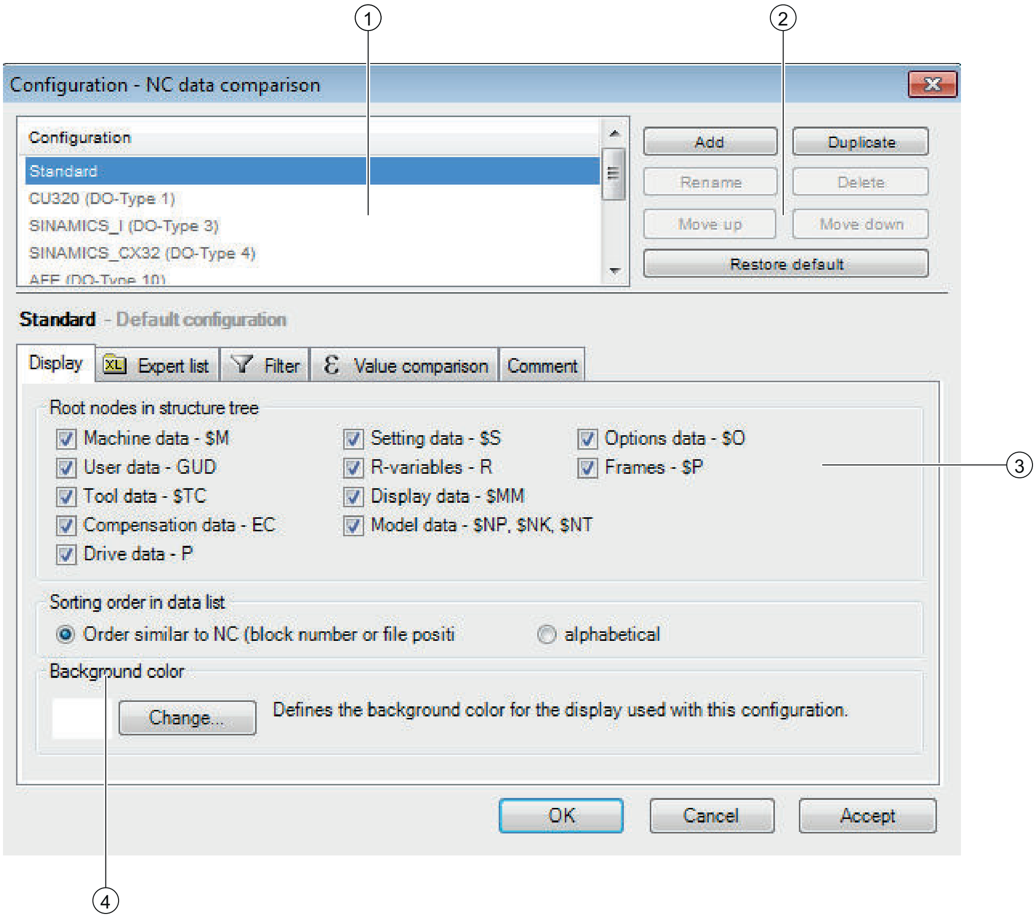
This information can only be completely displayed for archives that contain both NC data as well as drive data.

3.3.3 Configuration

NC data comparison

You can call the dialog for configuration of the NC data comparison via the "Comparison configuration" button ⚒. You can manage the different comparison configurations (i.e. create, edit, delete, etc.) in the dialog box.

In a comparison configuration, all of the criteria that should be taken into account when comparing NC data can be defined.
You can also export or import configurations, for example, to make them available to other users (see Exporting and importing configurations (Page 75)).



Configuration

Standard
CU320 (DO-Type 1)
SINAMICS_I (DO-Type 3)
SINAMICS_CX32 (DO-Type 4)
AFE (DO-Type 10)

Standard - Default configuration

Root nodes in structure tree

☑ Machine data - $M
☑ User data - GUD
☑ Tool data - $TC
☑ Compensation data - EC
☑ Drive data - P
☑ Setting data - $S
☑ R-variables - R
☑ Display data - $MM
☑ Model data - $NP, $NK, $NT
☑ Options data - $O
☑ Frames - $P

Sorting order in data list

● Order similar to NC (block number or file positi ○ alphabetical

Background color

Defines the background color for the display used with this configuration.

① List of the available comparison configurations

② Buttons for managing the configurations

③ The properties that belong to the selected configuration are saved in this area of the configuration dialog; these properties include display criteria, filter criteria and a comment.

④ Background color

① List of the available comparison configurations

The existing comparison configurations are listed in this area of the configuration dialog and can be selected by clicking on them.

Configurations shown in gray are user-specified configurations; these can be changed, but not deleted - and can be restored again at any time.

① Buttons for managing the configurations

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 user-specified setting", the changes to the specified comparison configuration are undone.

Note

The "Restore user-specified setting" 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.

① Background color

So that the user can immediately recognize how to select certain comparison configurations, a separate background color can be defined for each comparison configuration.

Display tab

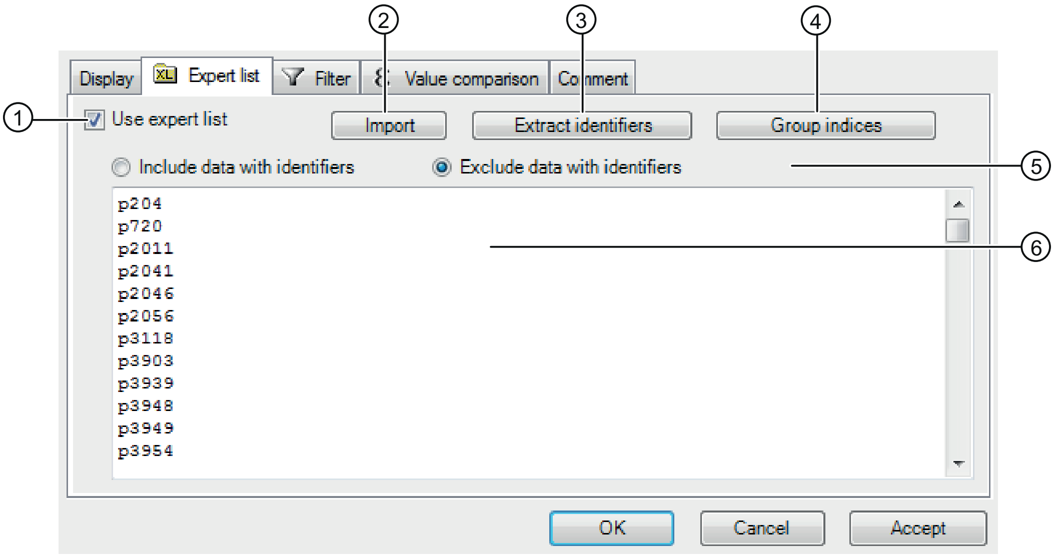
Under the "Display" tab, you can define which main nodes should be displayed in the data structure tree, assuming that data is available in the comparison object (archive or file) for this main node. Further, you can define whether the data should be sorted in alphabetic order or similar to that used in an NC in the data list.

Expert list tab

Using the "Expert list" tab, you can specify a (large) number of identifiers that should either be included or excluded for the comparison.
Contrary to the functionality for filters, the identifiers must be completely specified. The only exception is the field data in the identifier.

Example:

An entry "$MA_CTRLOUT_NR[0]" in an inclusion list precisely includes this identifier.
An entry "$MA_CTRLOUT_NR" in an inclusion list includes all identifiers that belong to this field, i.e. "$MA_CTRLOUT_NR[0]" and "$MA_CTRLOUT_NR[1]".



☑ Use expert list

○ Include data with identifiers ● Exclude data with identifiers

```
p204
p720
p2011
p2041
p2046
p2056
p3118
p3903
p3939
p3948
p3949
p3954
```

① Using the expert list
② **Import...** button
③ **Extracting identifiers** button
④ **Combining indices** button
⑤ Including/excluding data with identifiers
⑥ Field listing the identifiers

① Use expert list

The use of an expert list can be activated by selecting "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 following symbol XL appears in the status bar.

---

② Import...

Identifiers can be imported from a file using this button.

① Extracting identifiers

If identifiers were imported via the clipboard or a file, then values or additional information that are not identifiers can be included.

Identifiers can be extracted from the imported characters using this button.

The "Extract identifiers" function also removes the field data at the identifiers and eliminates identifiers that occur more than once in the list.

④ Combining indices

This button can be used to combine indexed machine data to form global machine data:

$MN_AXCONF_MACHAX_NAME_TAB[3] → $MN_AXCONF_MACHAX_NAME_TAB

① Include/exclude data with identifiers

You can define as to whether the listed identifiers and their data are included or excluded when making the comparison.

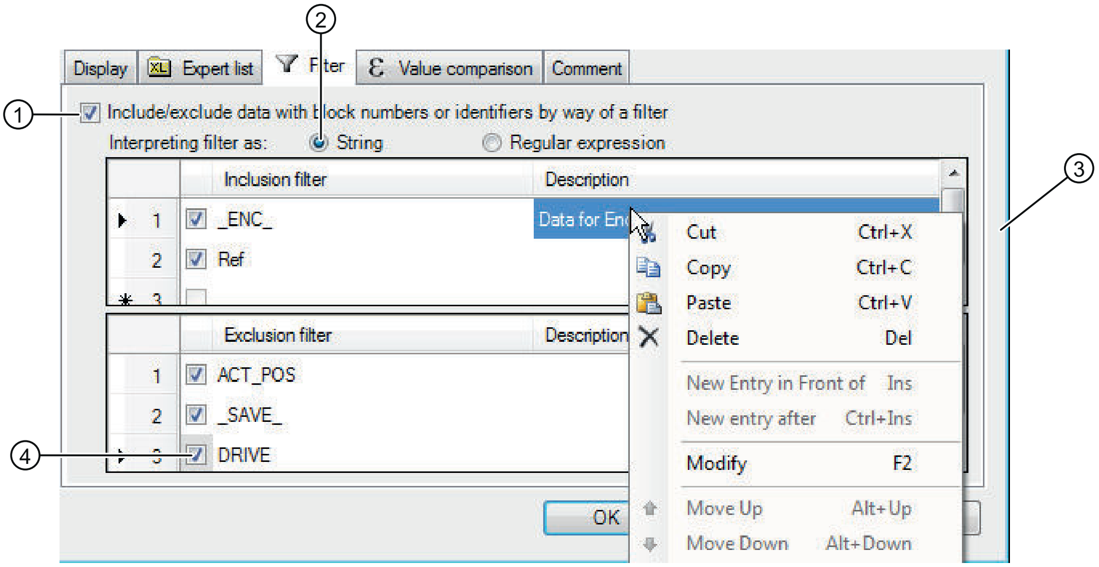
⑥ Field list of the identifiers

The identifiers involved are displayed in this field. There may only be one identifier in each line.

The identifiers can also be imported via the clipboard or using the import function.

Filter tab

Using filters, identifiers can be included or excluded for the comparison by identifying matching characters.
The filters are applied to the identifiers and their (block) number. The function is not case-sensitive.



☑ Include/exclude data with block numbers or identifiers by way of a filter

Interpreting filter as: ● String ○ Regular expression

		Inclusion filter	Description
▶	1	☑ _ENC_	Data for En
	2	☑ Ref	
*	3	□	

		Exclusion filter	Description
	1	☑ ACT_POS	
	2	☑ _SAVE_	
▶	3	☑ DRIVE	

① Activate filter

② Define filters, as character string or regular expression

③ Filter entries can be created, edited and deleted via the context menu.

④ Every filter entry can be individually activated or deactivated.

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 all of the identifiers that have been filtered-out are not displayed and are also not compared. The filter symbol appears in the status bar.

---

② Define filter, as character stringor regular expression

Interpret filter as:

You can define whether the filter should be interpreted as **String** or **Regular expression** (see Use of regular expressions in Diff (Page 79)).

Generally, the filters do not describe complete identifiers, but distinct character strings of the identifier which are sufficient to clearly identify the required quantity of identifiers.

③ Activate or deactivate an exclusion filter

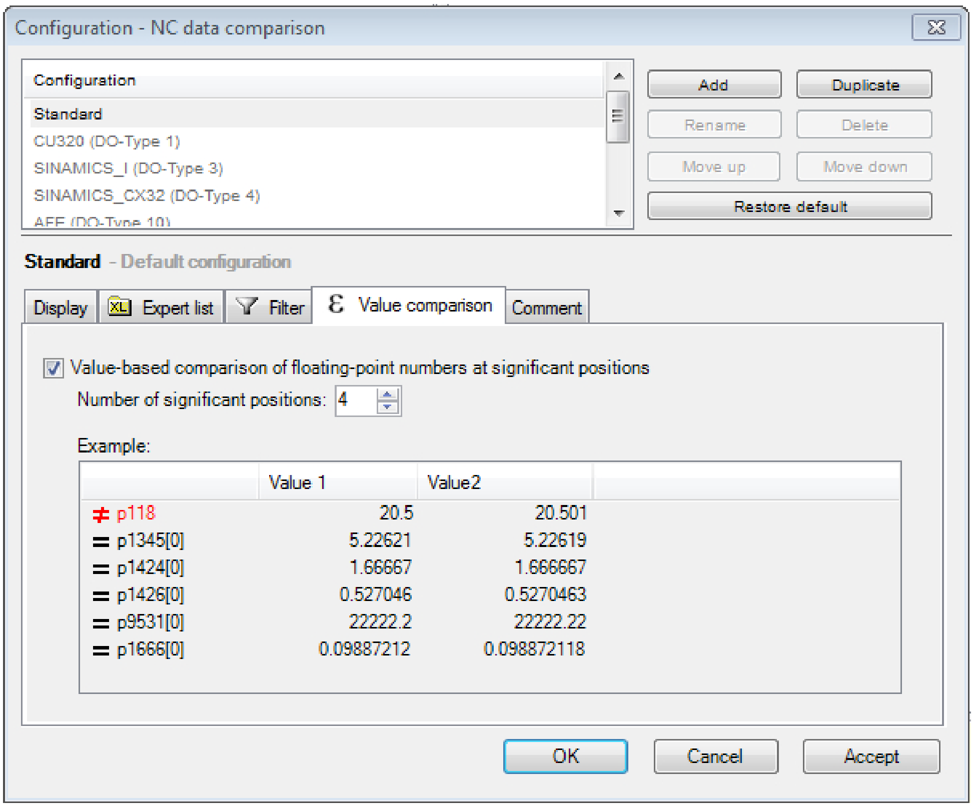
Every filter entry can be individually activated or deactivated.

Example of a filter:

SAFE	This filter describes all identifiers that contain the specified character string. Generally, these are all of the safety data such as $MA_SAFE_IS_ROT_AX, $MA_SAFE_SINGLE_ENC, etc.

Value range tab

In the Value range tab, the value-based comparison of floating point numbers is displayed at significant points.



Configuration - NC data comparison

Configuration
Standard
CU320 (DO-Type 1)
SINAMICS_I (DO-Type 3)
SINAMICS_CX32 (DO-Type 4)
AFE (DO-Type 10)

Standard - Default configuration

☑ Value-based comparison of floating-point numbers at significant positions
Number of significant positions: 4

Example:

	Value 1	Value2
≠ p118	20.5	20.501
= p1345[0]	5.22621	5.22619
= p1424[0]	1.66667	1.666667
= p1426[0]	0.527046	0.5270463
= p9531[0]	22222.2	22222.22
= p1666[0]	0.09887212	0.098872118

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 thus 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 on the selected comparison configuration can be saved under the Comment tab.

3.3.4 Copying/exporting data from the NC data comparison

In the NC data comparison, data identifiers and the associated values can be copied to the clipboard or exported to a file.

For instance, the copied or exported data can be used in manipulation jobs of Expert, in Excel for documentation, in another software application or where relevant, also in an NC data comparison in Diff.

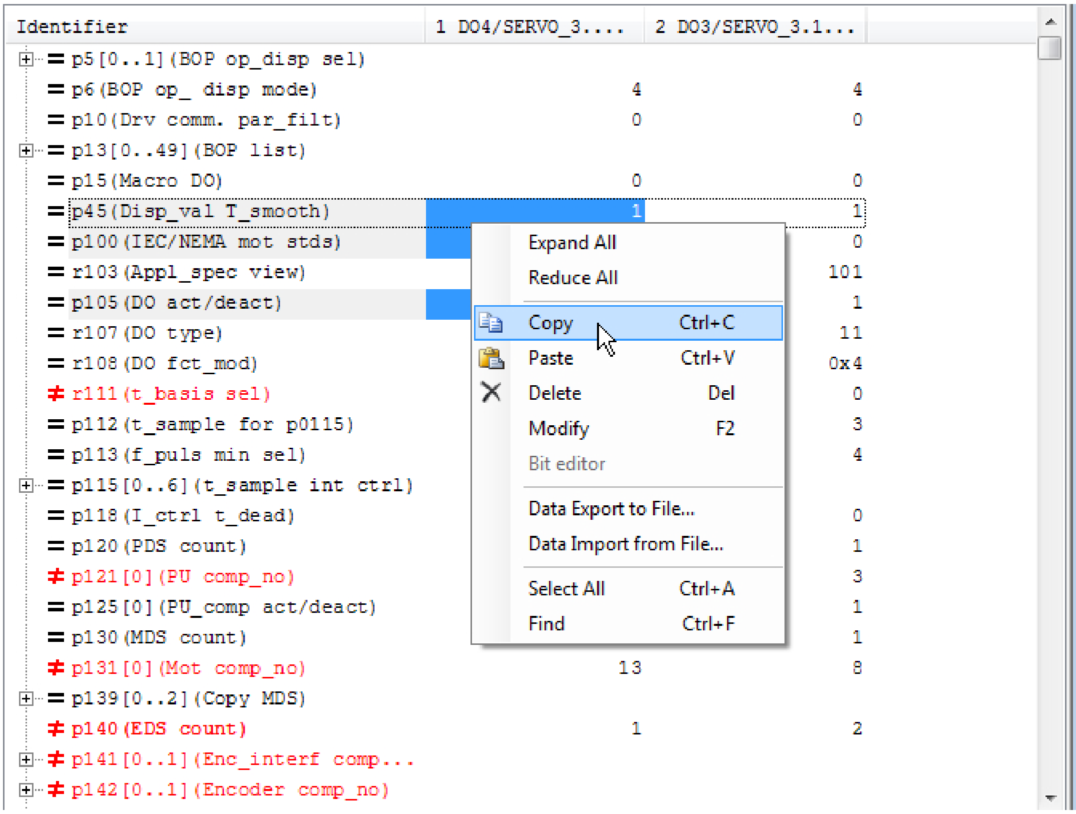
In the NC data comparison, depending on the data marking, there are two different copy and/or export formats "Value assignment" and "Value table".

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 are saved in a form that can be directly used in manipulation jobs of Expert.
This type of copy and export is the most frequently used type.

To copy or export data, the values must be selected in the value column as shown below.

Example 1: Copying of SINAMICS data



Identifier	1 DO4/SERVO_3....	2 DO3/SERVO_3.1...
= p5[0..1](BOP op_disp sel)		
= p6(BOP op_ disp mode)	4	4
= p10(Drv comm. par_filt)	0	0
= p13[0..49](BOP list)		
= p15(Macro DO)	0	0
= p45(Disp_val T_smooth)	1	1
= p100(IEC/NEMA mot stds)		0
= r103(Appl_spec view)		101
= p105(DO act/deact)		1
= r107(DO type)		11
= r108(DO fct_mod)		0x4
≠ r111(t_basis sel)		0
= p112(t_sample for p0115)		3
= p113(f_puls min sel)		4
= p115[0..6](t_sample int ctrl)		
= p118(I_ctrl t_dead)		0
= p120(PDS count)		1
≠ p121[0](PU comp_no)		3
= p125[0](PU_comp act/deact)		1
= p130(MDS count)		1
≠ p131[0](Mot comp_no)	13	8
= p139[0..2](Copy MDS)		
≠ p140(EDS count)	1	2
≠ p141[0..1](Enc_interf comp...		
≠ p142[0..1](Encoder comp_no)		

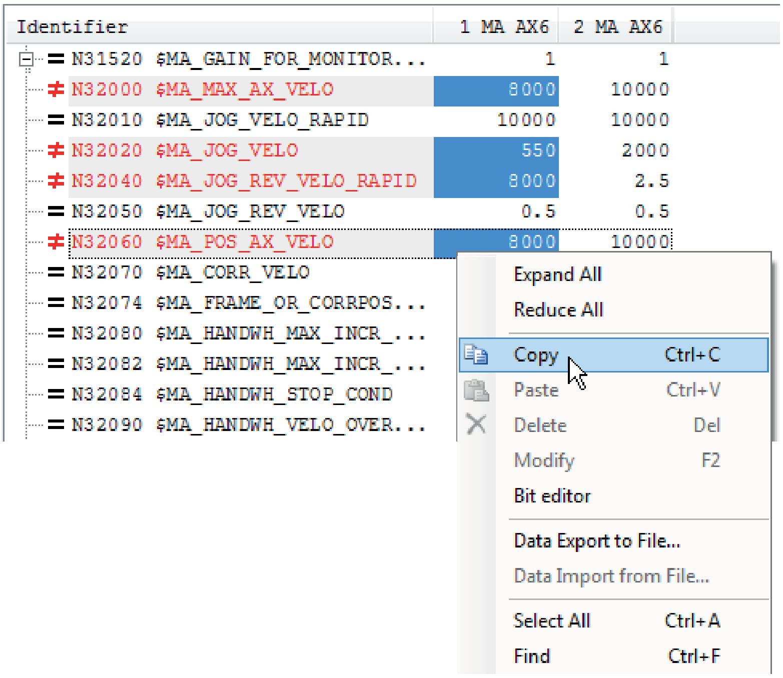
Using the context menu entries "Copy" or "Data Export to File", you can transfer the selected values and their identifiers to the clipboard or to a file.

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 a format that is in conformance with Expert, so that it can be used in a manipulation jobs 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 located in lines 2 to n.

Example 2: Copying NC axis machine data



Identifier	1 MA AX6	2 MA AX6
= N31520 $MA_GAIN_FOR_MONITOR...	1	1
≠ N32000 $MA_MAX_AX_VELO	8000	10000
= N32010 $MA_JOG_VELO_RAPID	10000	10000
≠ N32020 $MA_JOG_VELO	550	2000
≠ N32040 $MA_JOG_REV_VELO_RAPID	8000	2.5
= N32050 $MA_JOG_REV_VELO	0.5	0.5
≠ N32060 $MA_POS_AX_VELO	8000	10000
= N32070 $MA_CORR_VELO		
= N32074 $MA_FRAME_OR_CORRPOS...		
= N32080 $MA_HANDWH_MAX_INCR_...		
= N32082 $MA_HANDWH_MAX_INCR_...		
= N32084 $MA_HANDWH_STOP_COND		
= N32090 $MA_HANDWH_VELO_OVER...		

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.

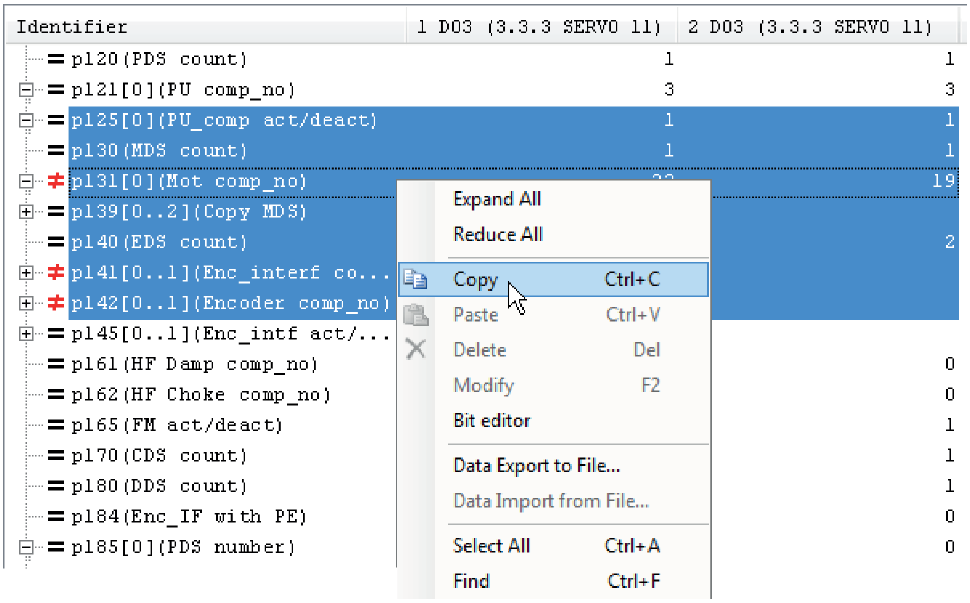
Note

CHANDATA(1) is also automatically added for channel-independent data as this guarantees that a previous channel assignment greater than 1 is reset again.
Channel-independent data must not be located below a CHANDATA operation 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 value of the particular comparison object (archive, file or node).

To copy or export data in this form, the identifiers in the Identifier column must be selected, as shown in the following example.



	Identifier	1 DO3 (3.3.3 SERVO 11)	2 DO3 (3.3.3 SERVO 11)
=	p120(PDS count)	1	1
=	p121[0](PU comp_no)	3	3
=	p125[0](PU_comp act/deact)	1	1
=	p130(MDS count)	1	1
≠	p131[0](Mot comp_no)		19
=	p139[0..2](Copy MDS)		
=	p140(EDS count)		2
≠	p141[0..1](Enc_interf co...		
≠	p142[0..1](Encoder comp_no)		
=	p145[0..1](Enc_intf act/...		
=	p161(HF Damp comp_no)		0
=	p162(HF Choke comp_no)		0
=	p165(FM act/deact)		1
=	p170(CDS count)		1
=	p180(DDS count)		1
=	p184(Enc_IF with PE)		0
=	p185[0](PDS number)		0

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
r131 [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

3.3.5 Pasting/importing data in the NC data comparison

Introduction

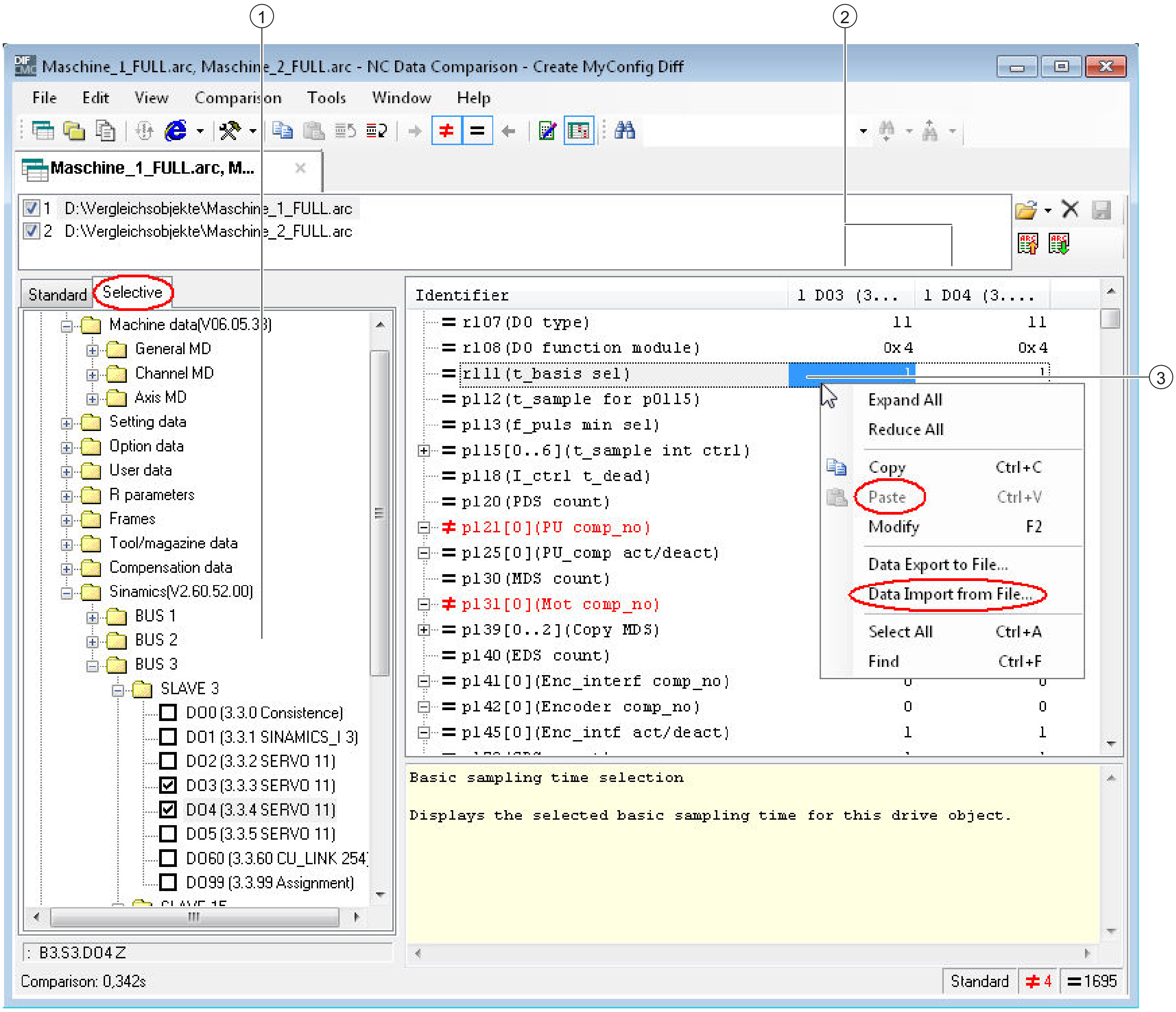
In the NC data comparison, data from the clipboard can be pasted or imported from a file. This function is **only** available in the "Selective" comparison mode and only for SINUMERIK archives (not INI/TEA files…).

The data must be 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 context menu it is possible to paste/import data to a value column ②.



Identifier	1 DO3 (3...	1 DO4 (3....
= r107(DO type)	11	11
= r108(DO function module)	0x4	0x4
= r111(t_basis sel)	1	1
= p112(t_sample for p0115)		
= p113(f_puls min sel)		
= p115[0..6](t_sample int ctrl)		
= p118(I_ctrl t_dead)		
= p120(PDS count)		
≠ p121[0](PU comp_no)		
= p125[0](PU_comp act/deact)		
= p130(MDS count)		
≠ p131[0](Mot comp_no)		
= p139[0..2](Copy MDS)		
= p140(EDS count)		
= p141[0](Enc_interf comp_no)	0	0
= p142[0](Encoder comp_no)	0	0
= p145[0](Enc_intf act/deact)	1	1

```
Basic sampling time selection

Displays the selected basic sampling time for this drive object.
```

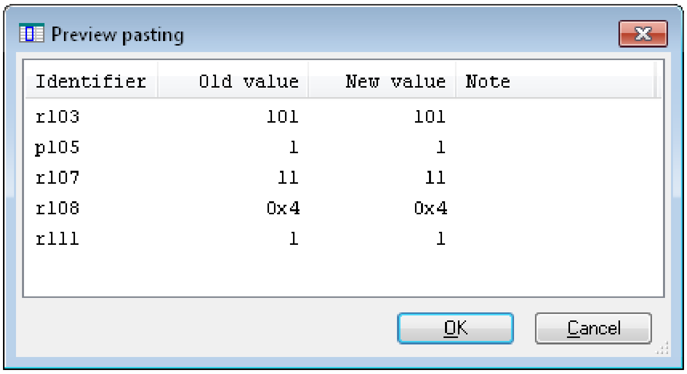
When pasting/importing data, Diff searches the identifiers contained in the clipboard/file in the data nodes ① selected in the value column ② and pastes the "new" value for the identifiers found.

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 a "Preview pasting" dialog.



Identifier	Old value	New value	Note
r103	101	101	
p105	1	1	
r107	11	11	
r108	0x4	0x4	
r111	1	1	

The following information is shown in the dialog:

- This dialog box lists all of the identifiers with the old and new values that are to be pasted. If, as a result of the new value, the value changes with respect to the old value, then the new value is shown in blue.
- If no identifiers (i.e. no old 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.

3.3.6 Data jobs in the NC data comparison

3.3.6.1 Overview

In the NC data 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, only the values of existing identifiers can be overwritten. Diff cannot insert any new identifiers in an archive.

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 NC data 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

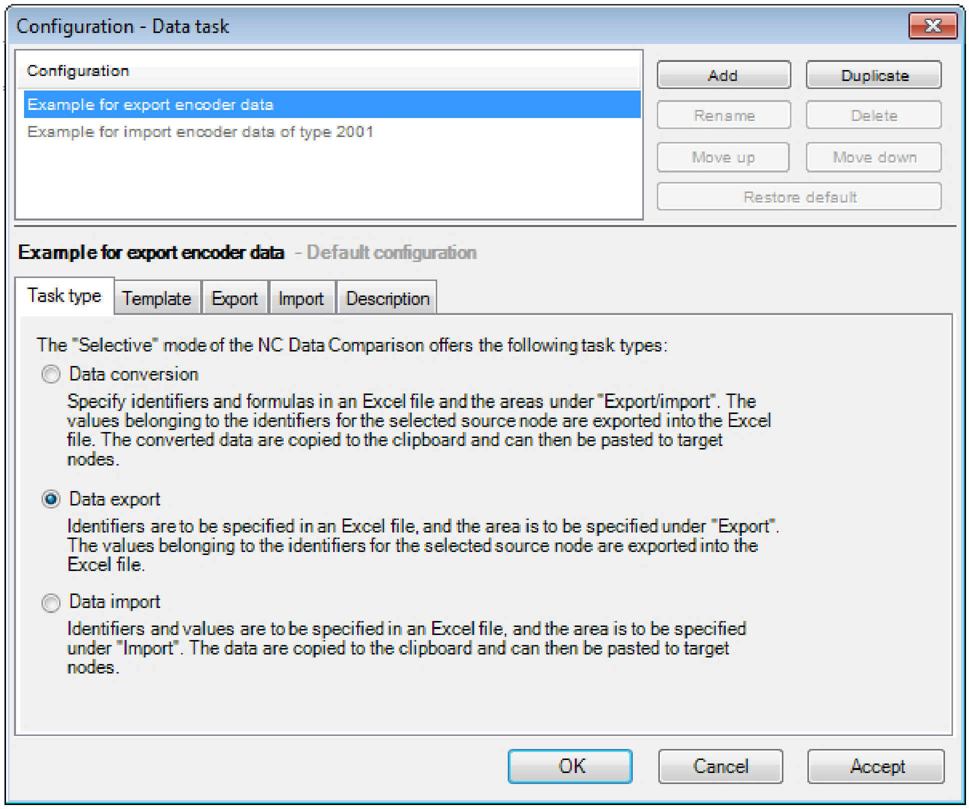
When executing conversion tasks, data identifiers are determined from an "Export area" of an Excel template. These identifiers are searched for in a selected data end node in the data structure of the NC data comparison and their values are transferred to an Excel template.
The conversion rule saved 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, only the values of existing identifiers can be overwritten. Diff cannot insert any new identifiers in an archive.
The Excel file with the converted data can be saved.

3.3.6.2 Configuring data jobs

You can call the dialog to configure data tasks via the main menu "Tools > NC Data Comparison > Configure Data Task".

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 provide them to other users (see Exporting and importing configurations (Page 75)).



Configuration - Data task

Configuration

Example for export encoder data

Example for import encoder data of type 2001

Example for export encoder data - Default configuration

The "Selective" mode of the NC Data Comparison offers the following task types:

○ Data conversion

Specify identifiers and formulas in an Excel file and the areas under "Export/import". The values belonging to the identifiers for the selected source node are exported into the Excel file. The converted data are copied to the clipboard and can then be pasted to target nodes.

● Data export

Identifiers are to be specified in an Excel file, and the area is to be specified under "Export". The values belonging to the identifiers for the selected source node are exported into the Excel file.

○ Data import

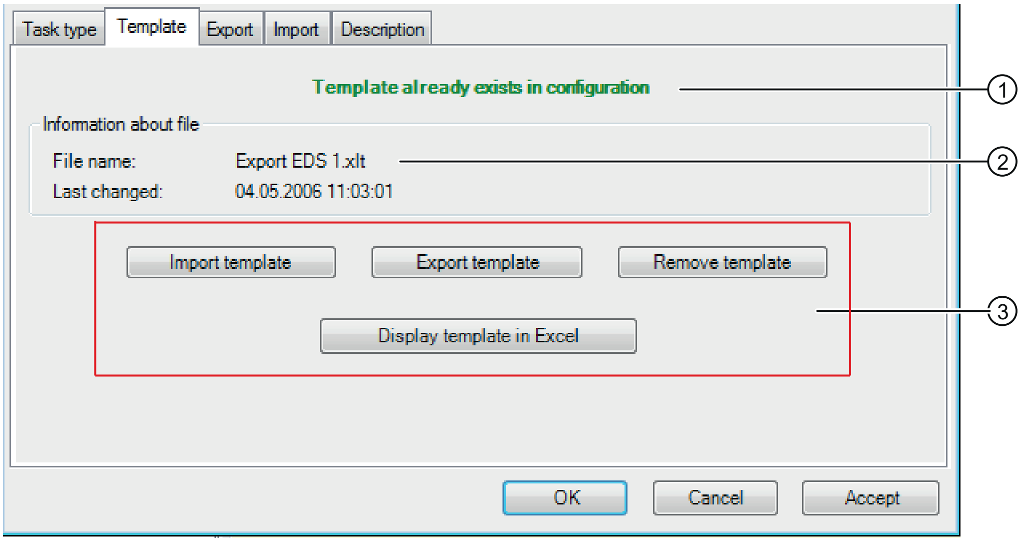
Identifiers and values are to be specified in an Excel file, and the area is to be specified under "Import". The data are copied to the clipboard and can then be pasted to target nodes.

Task type tab

Under the "Task type" tab, you can define whether a data task involves "Data conversion", "Data export" or "Data import".

Template tab

You can store an Excel template under 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 under this tab, then when executing the data task, Diff requests 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, which is assigned during the package execution.



① This inscription indicates that an Excel template is available in the configuration.

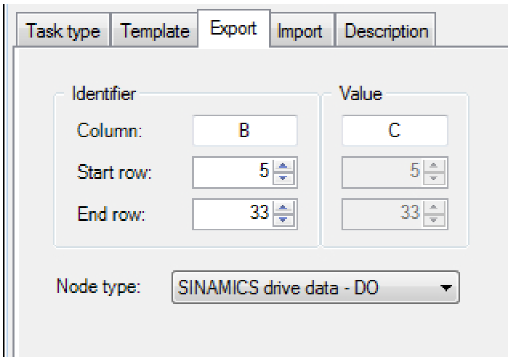
② In this field, the name and the date of the file are displayed, which should be used as Excel template.

③ Using these buttons, it is possible to

- 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 under the "Export" tab.



Identifier
Column: B
Start row: 5
End row: 33

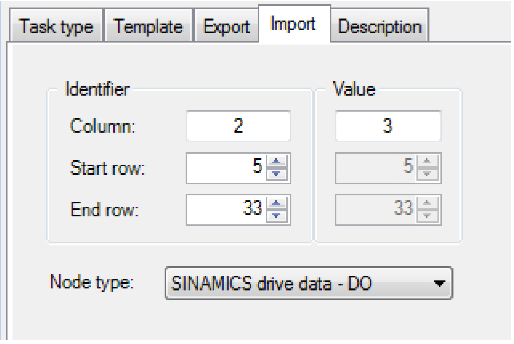
Value
Column: C
Start row: 5
End row: 33

Node type: SINAMICS drive data - DO

Further, a node type in the data structure tree must be specified under this tab. An export or conversion task can only be started at 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 under the "Import" tab.



Identifier
Column: 2
Start row: 5
End row: 33

Value
Column: 3
Start row: 5
End row: 33

Node type: SINAMICS drive data - DO

In addition, only a node type in the data structure tree must be specified for an import task under this tab. An import task can only be started at this node type. The node type should be selected that belongs to the data to be imported.

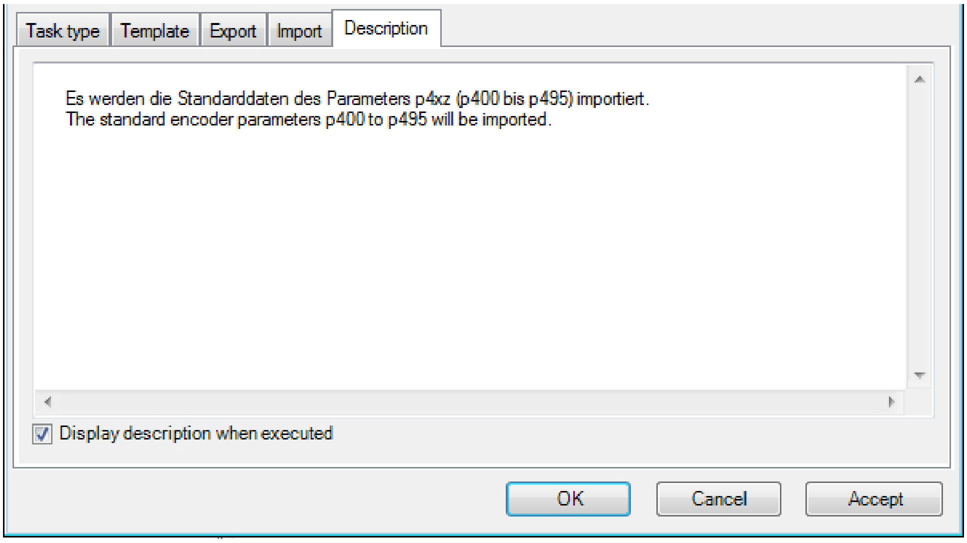
Note

For a conversion task, the selection of the node type under 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 can be saved under the "Description" tab.

The description is displayed in the data task wizard when the "Display description when executed" checkbox is activated.



Es werden die Standarddaten des Parameters p4xz (p400 bis p495) importiert.
The standard encoder parameters p400 to p495 will be imported.

☑ Display description when executed

3.3.6.3 Executing data tasks

Configured data tasks can be executed using a data task wizard.

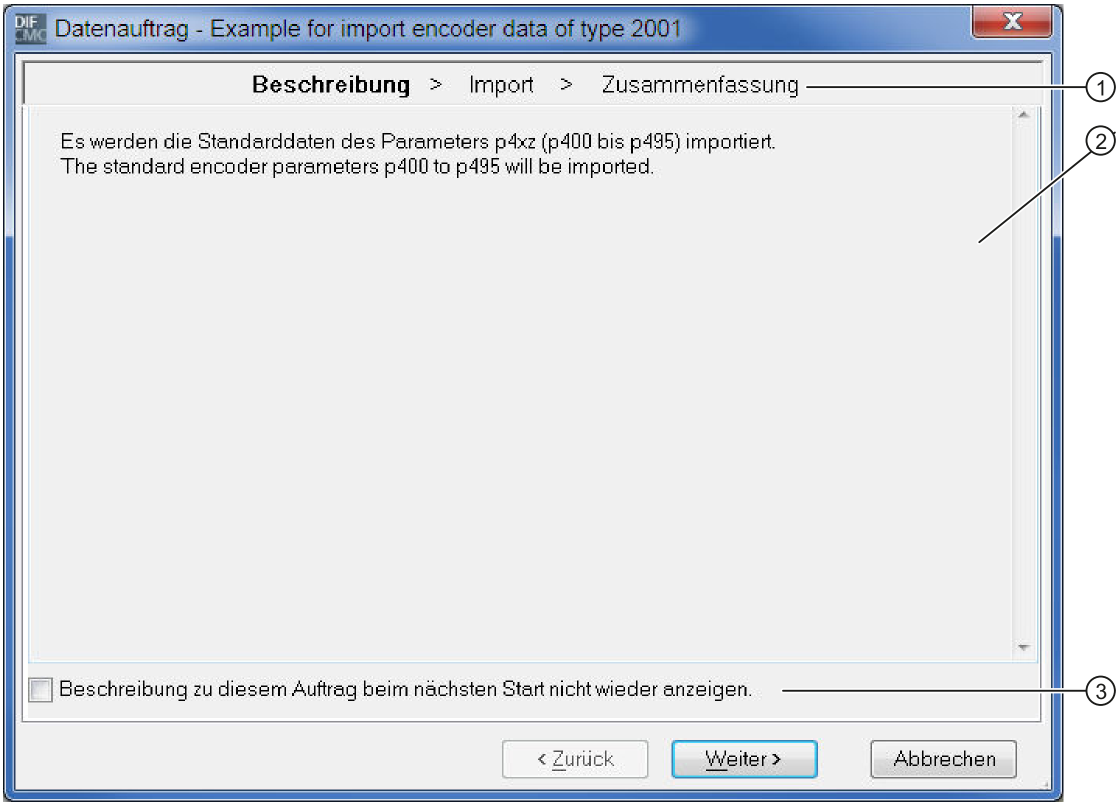
You can start a data task using the context menu at a selected data end node in the selective mode of the NC data comparison.

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 (see Configuring data jobs (Page 38)).

The data task wizard starts with the first dialog page after clicking on the required task. The dialogs 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.


Datenauftrag - Example for import encoder data of type 2001
Beschreibung > Import > Zusammenfassung
①
②
③
< Zurück
Weiter >
Abbrechen

Es werden die Standarddaten des Parameters p4xz (p400 bis p495) importiert.
The standard encoder parameters p400 to p495 will be imported.

□ Beschreibung zu diesem Auftrag beim nächsten Start nicht wieder anzeigen.

① **Identification of the active step**

The current step and the following or processed steps 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**

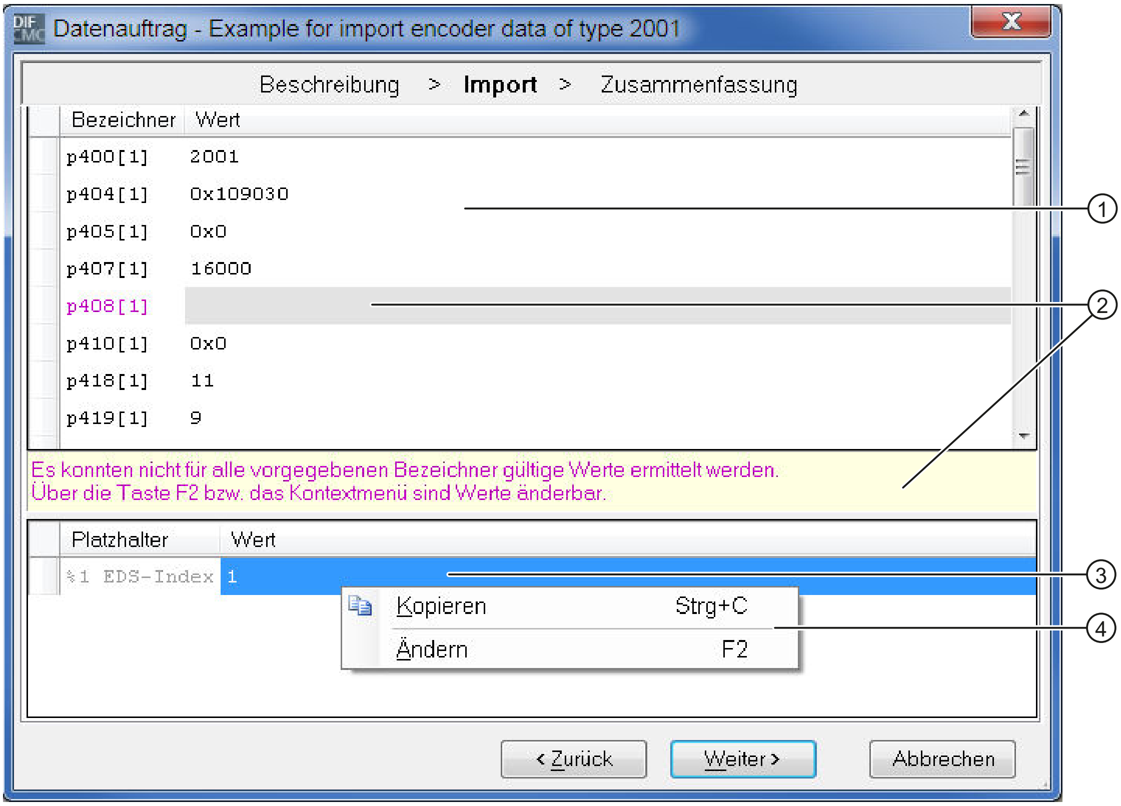
By checking the checkbox, you can select that the wizard no longer displays this dialog page the next time this data task is started.

Note

If the dialog is to be subsequently reactivated, then this is possible via the data task configuration (see Configuration of data tasks (Page 38)).

Dialog page: Import

The "Import" dialog page allows the import values - determined by the wizard from the Excel template - to be viewed.



Datenauftrag - Example for import encoder data of type 2001

Bezeichner	Wert
p400[1]	2001
p404[1]	0x109030
p405[1]	0x0
p407[1]	16000
p408[1]	
p410[1]	0x0
p418[1]	11
p419[1]	9

Es konnten nicht für alle vorgegebenen Bezeichner gültige Werte ermittelt werden.
Über die Taste F2 bzw. das Kontextmenü sind Werte änderbar.

Platzhalter	Wert
%1 EDS-Index	1

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 **Note on creating missing values**

If no values were able to be determined in the Excel template for individual identifiers, then these identifiers are color-coded and an information text is generated.

Values can be manually entered or changed for the identifiers; however, identifiers without value can also be processed.

③ **List of placeholders/values**

The place holders defined in the Excel template and their (default) values are listed.

④ **Change of the values via the context menu**

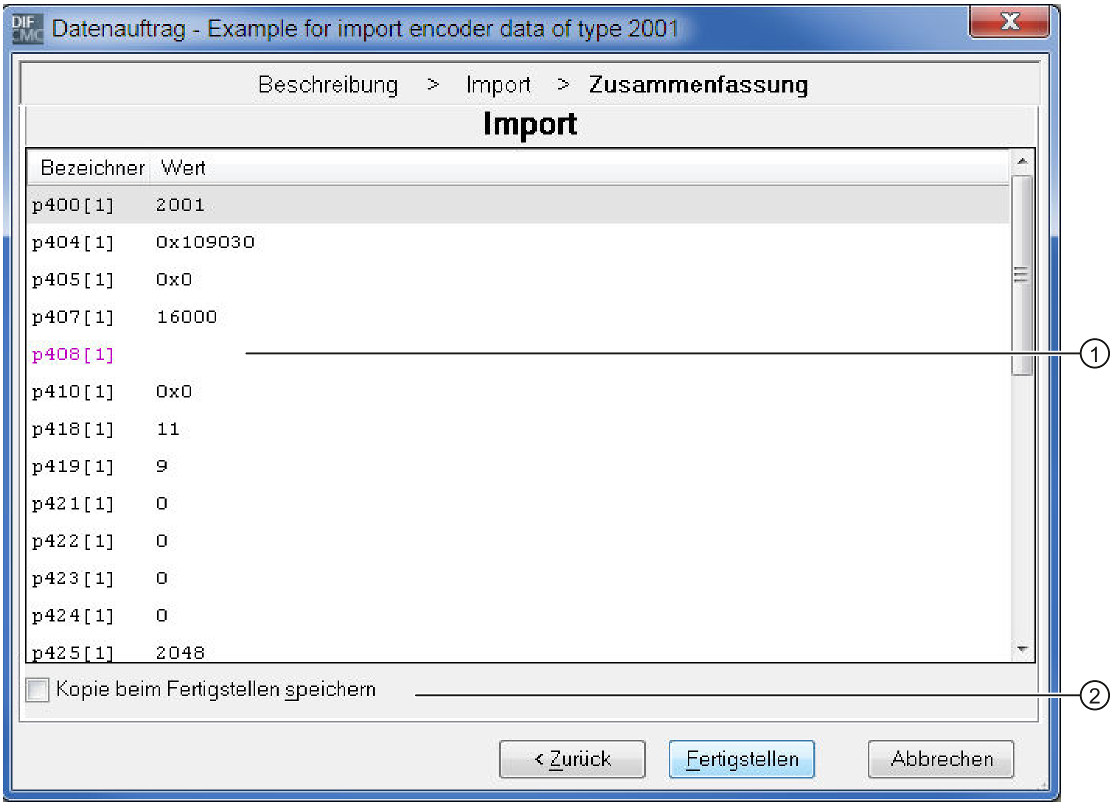
The default value saved for the place holder can be changed via the context menu.

If a place holder was defined in the Excel template, however no default value, then the wizard displays a question mark '?' instead of the value at ③ and in the identifiers at ① and a corresponding information text is displayed in area ③.

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.


Datenauftrag - Example for import encoder data of type 2001
Beschreibung > Import > Zusammenfassung
Import
①
②
☐ Kopie beim Fertigstellen speichern
< Zurück
Fertigstellen
Abbrechen

Bezeichner	Wert
p400[1]	2001
p404[1]	0x109030
p405[1]	0x0
p407[1]	16000
p408[1]	
p410[1]	0x0
p418[1]	11
p419[1]	9
p421[1]	0
p422[1]	0
p423[1]	0
p424[1]	0
p425[1]	2048

① **Color coding of the values**

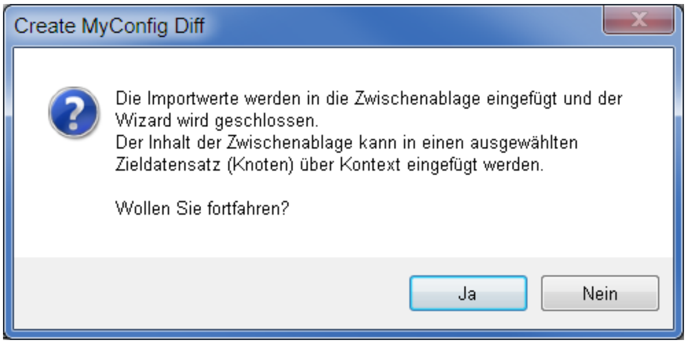
If, in the Excel template, no values were able to be determined for the individual identifiers, then these identifiers are shown in a bright purple color.
These values are shown in blue if the user manually entered or manipulated the values.

② **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.

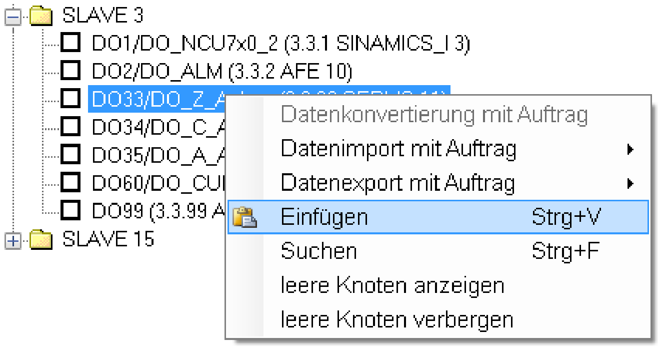
A dialog is then called which makes reference to the fact that the import values were transferred into the clipboard and can be pasted at a data node.



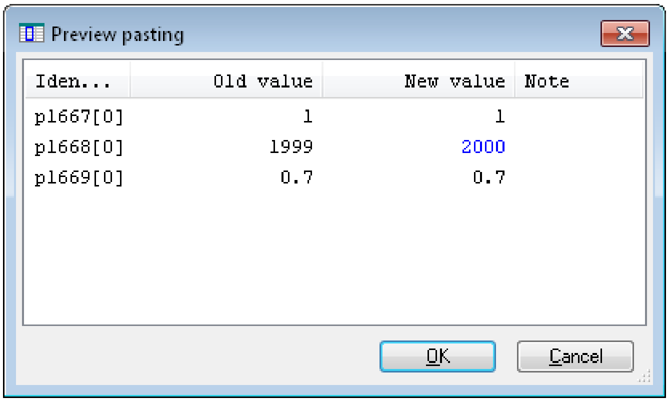
Die Importwerte werden in die Zwischenablage eingefügt und der Wizard wird geschlossen.
Der Inhalt der Zwischenablage kann in einen ausgewählten Zieldatensatz (Knoten) über Kontext eingefügt werden.

Wollen Sie fortfahren?

After clicking the "Yes" button, the converted data can be pasted at a data node via the context menu.



The "Preview pasting" dialog is displayed; this shows the value changes after the pasting.



Iden...	Old value	New value	Note
p1667[0]	1	1	
p1668[0]	1999	2000	
p1669[0]	0.7	0.7	

Pasting can be acknowledged – i.e. executed – by clicking the "OK" button. The "Cancel" button can be used to cancel the operation.

Values which change as a result of the pasting are shown in blue.
Users should take special note of values or elements shown in red – these are also allocated an information text.

3.4 Folder comparison

3.4.1 General information

In a Folder comparison of Diff, you can compare two comparison objects with each other, insert or delete files or folders, and store the modified objects again. The comparison and display of the comparison results is realized at the folders and files level. This means that the contents of the files are compared optionally, but not displayed.

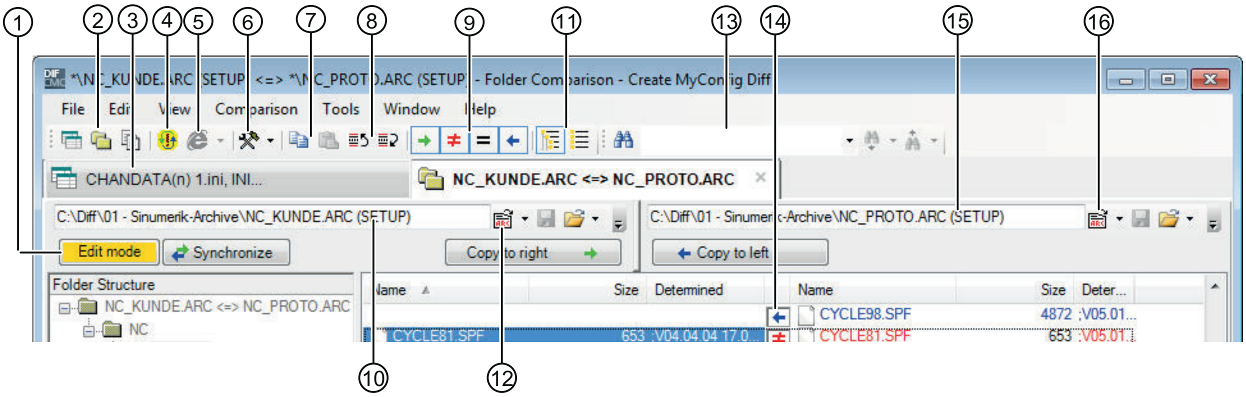
The folder comparison supports comparison and editing of the following objects:

- Folders/directories,
- SINUMERIK archive in the PC format
- SINUMERIK archive in the punched tape format
- ZIP archive
- TGZ archive
- TAR archive

Because unpacked archives usually 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 contents.
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.).

3.4.2 Overview of user interface

3.4.2.1 Folder comparison - toolbar


①
②
③
④
⑤
⑥
⑦
⑧
⑨
⑪
⑬
⑭
⑮
⑯
*\NC_KUNDE.ARC (SETUP) <=> *\NC_PROTO.ARC (SETUP) - Folder Comparison - Create MyConfig Diff
File
Edit
View
Comparison
Tools
Window
Help
CHANDATA(n) 1.ini, INI...
NC_KUNDE.ARC <=> NC_PROTO.ARC
C:\Diff\01 - Sinumerik-Archive\NC_KUNDE.ARC (SETUP)
C:\Diff\01 - Sinumerik-Archive\NC_PROTO.ARC (SETUP)
Edit mode
Synchronize
Copy to right
Copy to left
Folder Structure
NC_KUNDE.ARC <=> NC_PROTO.ARC
NC
Name
Size
Determined
Name
Size
Deter...
CYCLE98.SPF
4872
;V05.01...
CYCLE81.SPF
653
;V04.04.04 17.0...
CYCLE81.SPF
653
;V05.01...
⑩
⑫

① Edit mode - Synchronize - Copy

② New folder comparison
Calling a folder comparison.

③ Switch between opened comparisons

④ Update comparison

⑤ Display/save HTML/XML report

⑥ Comparison configuration

⑦ Copy / Insert

⑧
Go to the next/previous difference.

⑨ Button group comparison results

⑩ Path of the opened comparison object.

⑪ Toggling between a tree view and list mode.

⑫ Selecting the **left-hand** comparison object.

⑬ Buttons/input for searching

⑭ Column with the comparison result symbols

⑮ Path of the opened comparison object.

⑯ Selecting the **right-hand** comparison object.
⇒ see ⑫

Description of buttons/fields of the toolbar

The buttons of the toolbar are described in more detail below.

① Edit mode - Synchronize - Copy

The edit mode can be activated and deactivated using the "Edit mode" button.

When the edit mode is activated, using the context menu, the following edit functions can be applied: Files and folders can be cut, copied, pasted, deleted and renamed. New files and folders can also be created.

The changes are saved in an archive using the floppy disk symbol.

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 symbol remains deactivated.

The content of the left-hand comparison object can be synchronized with the content of the right-hand comparison object using the "Synchronize" button.

You can select which files and folders should be adapted by changing the comparison symbols (see ⑭) with a left click or from the short-circuit menu.

With the "Copy to the right," the selected files and folders of the left-hand comparison object are inserted in the right-hand comparison object.

With the "Copy to the left," the selected files and folders of the right-hand comparison object are inserted in the left-hand comparison object.

③ Switch between opened comparisons

In the opened tabs, similar to the Window task bar, you can toggle between the opened comparisons with a mouse click or by simultaneously pressing "Ctrl" and "Tab".

④ Update comparison

The update button is always shown in color if the displayed comparison result is not a current result and the comparison should be restarted using this button.

① Display/save HTML/XML report

The comparison result of the nodes selected in the folder structure is shown in the standard browser (Internet Explorer) using this button.

The result can be saved either as HTML or XML protocol.

The comparison result for the selected directory is output completely or, under "...like display", an output depending on the settings made under ⑨.

⑥ Comparison configuration

Using this button, a comparison configuration is changed or a different comparison configuration is selected (see Configuration (Page 55)).

⑥ Copy / Insert

Buttons for copying and inserting (only in edit mode) selected folders or files of the left-hand or right-hand comparison object (see Copy from the folder comparison (Page 59)).

⑨ Comparison results button group

The associated lines of the comparison result can be hidden or displayed using this group of buttons.

① Toggle between a tree view and list mode.

Improved synchronization functionality is obtained by being able to toggle.

⑫ Selecting the left-hand comparison object.

Selection of the folder to be compared
Selection of the archive to be compared
Selection from the objects that were last selected
Load archive from the control
Load archive into the control

⑬ Buttons/input for searching

This group contains:

- a button for opening a dialog box in which the search options can be defined
- an edit box for searching for character strings in selected nodes
- buttons for searching upward and downward.

Search down
Search up

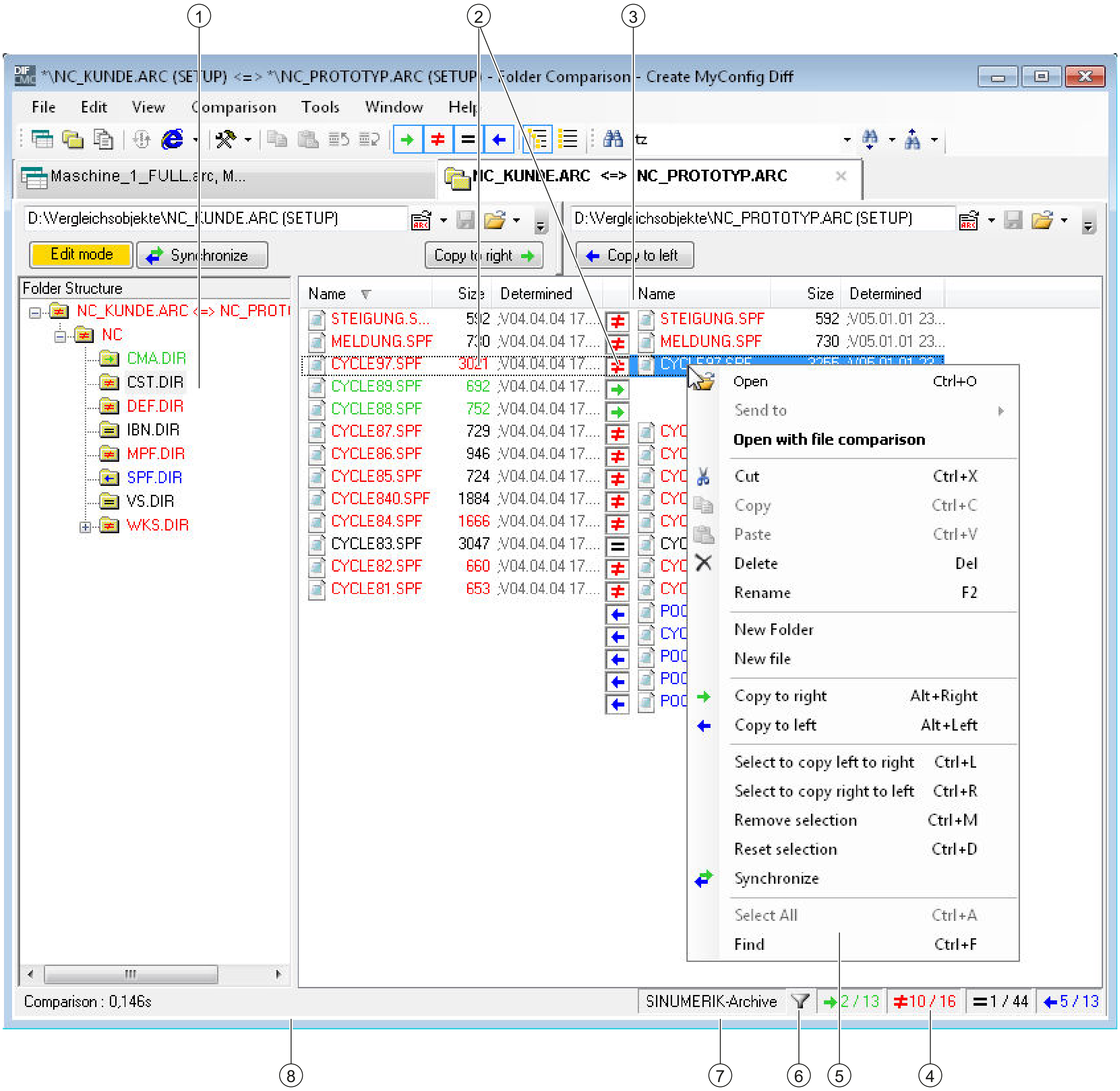
⑭ Column with the comparison result symbols

For separating the left-hand and right-hand comparison objects.

When the edit mode is activated ①, the symbols can be changed for the purpose of synchronization (left click, context menu).

3.4.2.2 Folder comparison - overview window

Below is an overview of the functions in the folder comparison window.



Name	Size	Determined		Name	Size	Determined
STEIGUNG.S...	592	V04.04 17....	≠	STEIGUNG.SPF	592	V05.01.01 23...
MELDUNG.SPF	730	V04.04 17....	≠	MELDUNG.SPF	730	V05.01.01 23...
CYCLE97.SPF	3021	V04.04 17....	≠	CYCLE97.SPF	2255	V05.01.01 23...
CYCLE89.SPF	692	V04.04 17....	→			
CYCLE88.SPF	752	V04.04 17....	→			
CYCLE87.SPF	729	V04.04 17....	≠	CYC		
CYCLE86.SPF	946	V04.04 17....	≠	CYC		
CYCLE85.SPF	724	V04.04 17....	≠	CYC		
CYCLE840.SPF	1884	V04.04 17....	≠	CYC		
CYCLE84.SPF	1666	V04.04 17....	≠	CYC		
CYCLE83.SPF	3047	V04.04 17....	=	CYC		
CYCLE82.SPF	660	V04.04 17....	≠	CYC		
CYCLE81.SPF	653	V04.04 17....	≠	CYC		
			←	POC		
			←	CYC		
			←	POC		
			←	POC		
			←	POC		

① Display of the folders with respective 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 Context menu

⑥ Comparison configuration

⑦ Display of the current comparison configuration (status bar)

⑧ Status bar

Description of the areas/functionalities in the Folder comparison window

The individual functionalities of the folder comparison are described in more detail below:

① Display of the folders with the respective 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 symbols.

② Colored comparison of the comparison objects

The content of folder ① selected in the area is displayed next to one another for both comparison objects, whereby the center column ③ with the comparison result symbols is also used as demarcation 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 on the column header.

Note

The contents of the files are only compared in the binary notation if this has been preselected in the comparison configuration (see Configuration (Page 55)).

The date for the comparison is not relevant for folders.

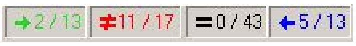
① Column of the comparison result icons

Column with the comparison result symbols, to separate the left-hand and right-hand comparison objects.
When the edit mode is activated, you can change the symbols for synchronization purposes by left-clicking or via the context menu.

Meaning of the comparison symbols:

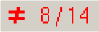
→ A green arrow indicates that the folder is only available in the left-hand comparison object.
≠ The unequal character indicates that the contents of the folder are not identical.
= The equal character indicates that the contents of both folders are identical.
← A blue arrow indicates that the folder is only available in the right-hand comparison object.

④ Assignment of comparison result icon/value (status bar bottom right)


→2 / 13 ≠11 / 17 =0 / 43 ←5 / 13

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:



≠ 8/14

Specifies that there are 8 unequal files from a total of 14 in the currently selected folder.

⑧ Special features context 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 Comparison configuration

A filter is used in the comparison configuration (see Configuration (Page 55)).

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.

① Display of the current comparison configuration (status bar)

The comparison configuration that is being used in the actual comparison is displayed in this field (see Configuration (Page 55)).

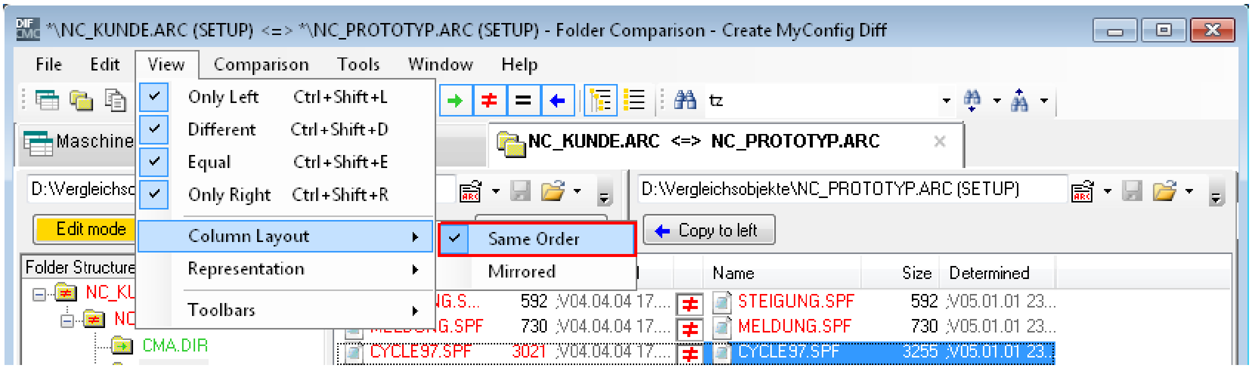
The comparison configuration is opened with a double-click.

3.4.3 Changing the column arrangement

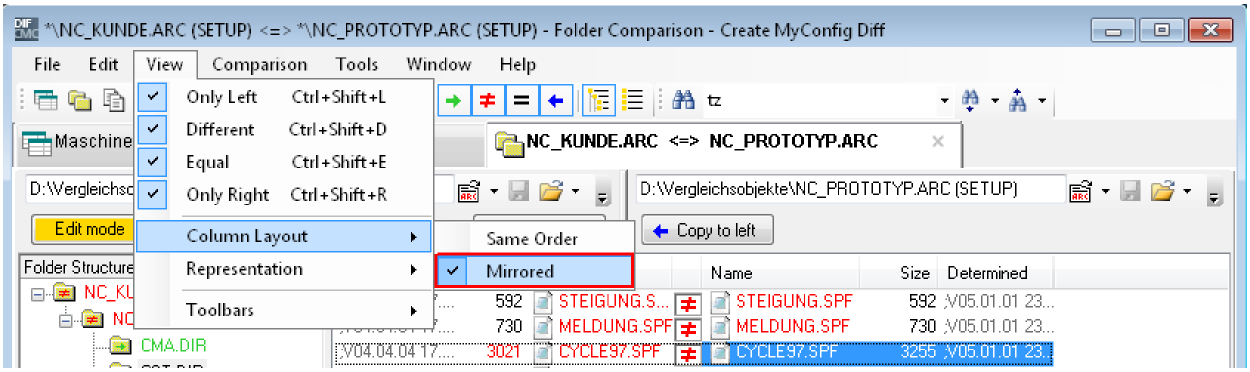
Changing the column arrangement for folder comparison

The column arrangement when comparing folders can be modified via the menu "View > "Column Arrangement" > "Same Sequence" or "Mirrored".

Same sequence


*\NC_KUNDE.ARC (SETUP) <=> *\NC_PROTOTYP.ARC (SETUP) - Folder Comparison - Create MyConfig Diff
File
Edit
View
Comparison
Tools
Window
Help
Only Left Ctrl+Shift+L
Different Ctrl+Shift+D
Equal Ctrl+Shift+E
Only Right Ctrl+Shift+R
Column Layout
Representation
Toolbars
Same Order
Mirrored
tz
Maschine
NC_KUNDE.ARC <=> NC_PROTOTYP.ARC
D:\Vergleichsc
D:\Vergleichsobjekte\NC_PROTOTYP.ARC (SETUP)
Edit mode
Copy to left
Folder Structure
NC_KU
NC
CMA.DIR
Name
Size
Determined
NG.S... 592 ;V04.04.04 17....
G.SPF 730 ;V04.04.04 17....
CYCLE97.SPF 3021 ;V04.04.04 17....
STEIGUNG.SPF 592 ;V05.01.01 23...
MELDUNG.SPF 730 ;V05.01.01 23...
CYCLE97.SPF 3255 ;V05.01.01 23...

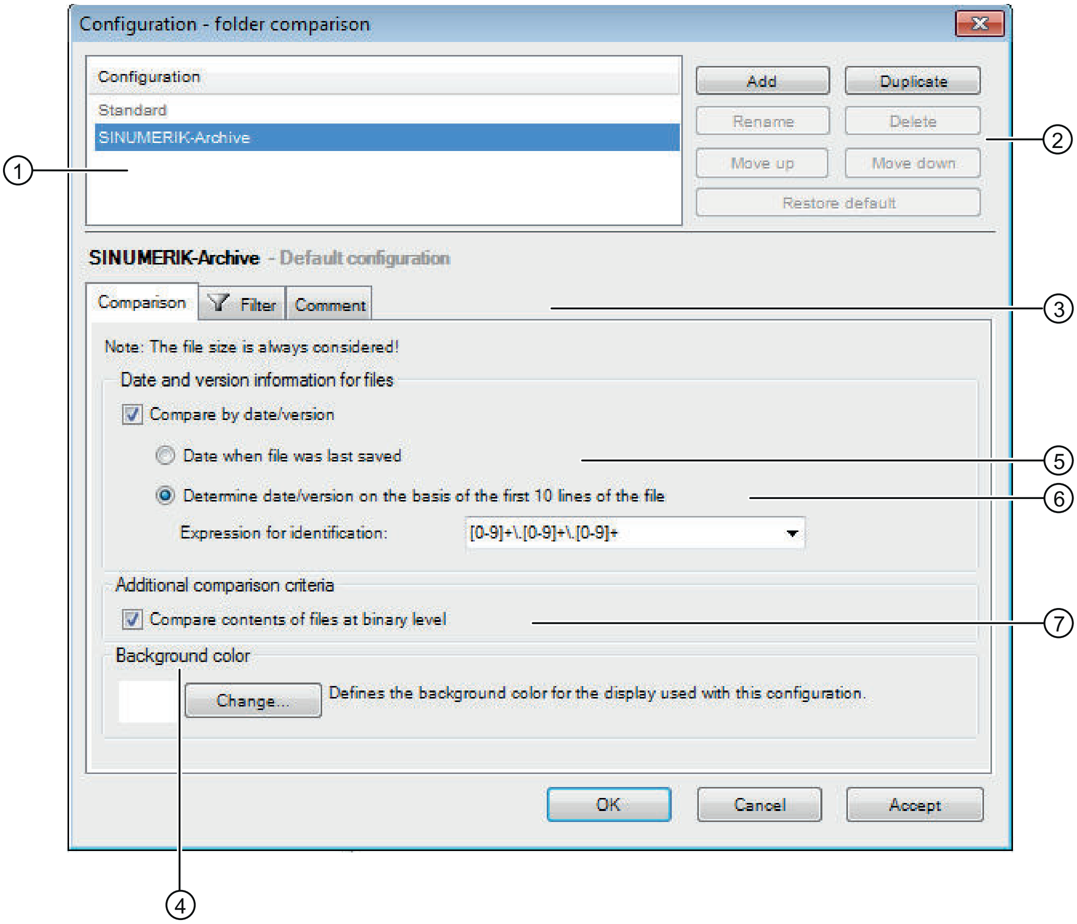
Mirrored


*\NC_KUNDE.ARC (SETUP) <=> *\NC_PROTOTYP.ARC (SETUP) - Folder Comparison - Create MyConfig Diff
File
Edit
View
Comparison
Tools
Window
Help
Only Left Ctrl+Shift+L
Different Ctrl+Shift+D
Equal Ctrl+Shift+E
Only Right Ctrl+Shift+R
Column Layout
Representation
Toolbars
Same Order
Mirrored
tz
Maschine
NC_KUNDE.ARC <=> NC_PROTOTYP.ARC
D:\Vergleichsc
D:\Vergleichsobjekte\NC_PROTOTYP.ARC (SETUP)
Edit mode
Copy to left
Folder Structure
NC_KU
NC
CMA.DIR
Name
Size
Determined
592 STEIGUNG.S...
730 MELDUNG.SPF
;V04.04.04 17.... 3021 CYCLE97.SPF
STEIGUNG.SPF 592 ;V05.01.01 23...
MELDUNG.SPF 730 ;V05.01.01 23...
CYCLE97.SPF 3255 ;V05.01.01 23...

3.4.4 Configuration

Folder comparison

You can call the dialog to configure a folder comparison using the "Comparison configuration" button. You can manage the different comparison configurations (i.e. create, edit, delete, etc.) in the dialog box.
In a comparison configuration, all of the criteria that should be taken into account when comparing folders can be defined.
You can also export or import configurations, for example, to make them available to other users (see Exporting and importing configurations (Page 75)).



Configuration
Standard
SINUMERIK-Archive

SINUMERIK-Archive - Default configuration

Note: The file size is always considered!

Date and version information for files

☑ Compare by date/version

○ Date when file was last saved

● Determine date/version on the basis of the first 10 lines of the file

Expression for identification: [0-9]+\.[0-9]+\.[0-9]+

Additional comparison criteria

☑ Compare contents of files at binary level

Background color

Defines the background color for the display used with this configuration.

① List of the comparison configurations
② Buttons for managing the configurations
③ Comparison / Filter / Comment tab
④ Background color
⑤ Date of the last change option button
⑥ Date field/version/drop-down list of regular identification expressions option button
⑦ Checkbox Additional comparison criteria

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 on them.

Configurations shown in gray are user-specified configurations; these can be changed, but not deleted - and can be restored again at any time.

① Buttons for managing the configurations

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 user-specified setting", the changes to the specified comparison configuration are undone.

Note

The "Restore user-specified setting" 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 / Filter / Comment tab

The properties that belong to the selected configuration are stored in this area of the Configuration dialog; these properties include comparison criteria, filter criteria and a comment.

① Background color

So that the user can immediately recognize how to select certain comparison configurations, a separate background color can be defined for each comparison configuration.

Comparison tab

When making a comparison, the file size is always taken into account and when it is equal, it is shown in black - and when it is unequal, in red.

⑤ Date of last change to file option button

When the date/version comparison is activated, then for this selection, Diff also takes into account the date that the file was last changed and displays this in either in black or red.

The date is not relevant for folders.

⑥ Date field/version/drop-down list of regular identification expression option button

SINUMERIK archives can contain cycles.

The cycle version number is specified in the file. When making a comparison, it provides more information than the date and can be used for the comparison.
In order to be able to compare these version numbers, the files in the first ten 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 archive".

Some regular expressions can be created in the drop down list box. An introduction to regular expressions can be found at the end of the document (see Use of regular expressions in Diff (Page 79)).

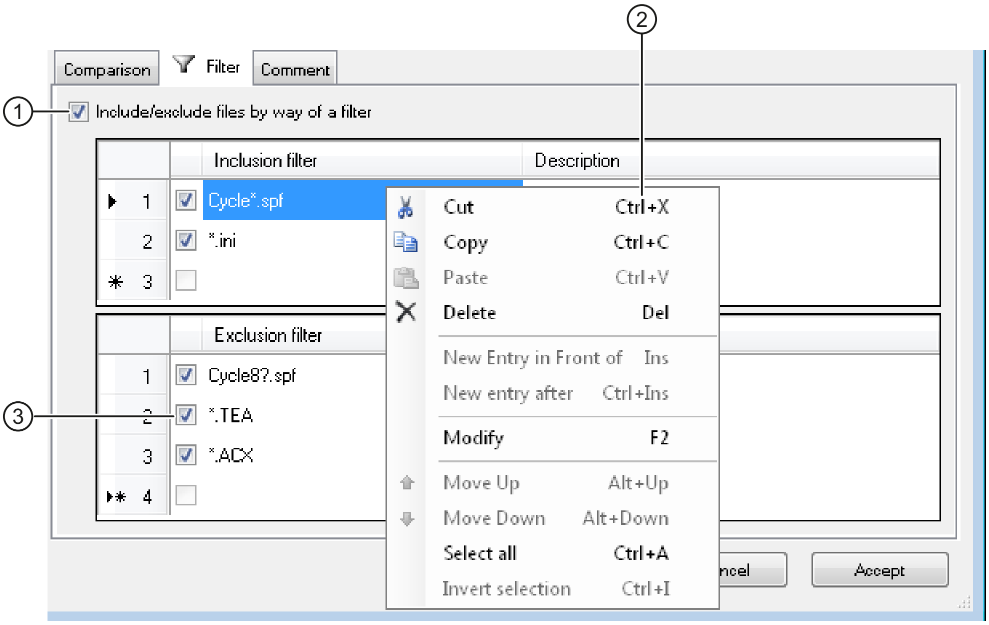
⑦ Additional comparison criteria checkbox

The content of the files is only compared byte-by-byte if the additional comparison criterion "Compare content of the files in a binary fashion" is selected.

Otherwise, only the size is compared and possibly the date or the version.

Filter tab

Using filters, files can be hidden and displayed for comparison in screen forms.



☑ Include/exclude files by way of a filter

		Inclusion filter	Description
▶ 1	☑	Cycle*.spf	
2	☑	*.ini	
✳ 3	□		

		Exclusion filter
1	☑	Cycle8?.spf
2	☑	*.TEA
3	☑	*.ACX
▶✳ 4	□	

① The functionality of the inclusion and exclusion filter is activated via the "Include/exclude files through filter" checkbox.
NOTE:
The filter symbol is shown in the status bar if this function is activated.

② Filter entries can be created, edited and deleted via the context menu.

③ Every filter entry can be individually activated or deactivated.

Table 3-1 Examples of filters:

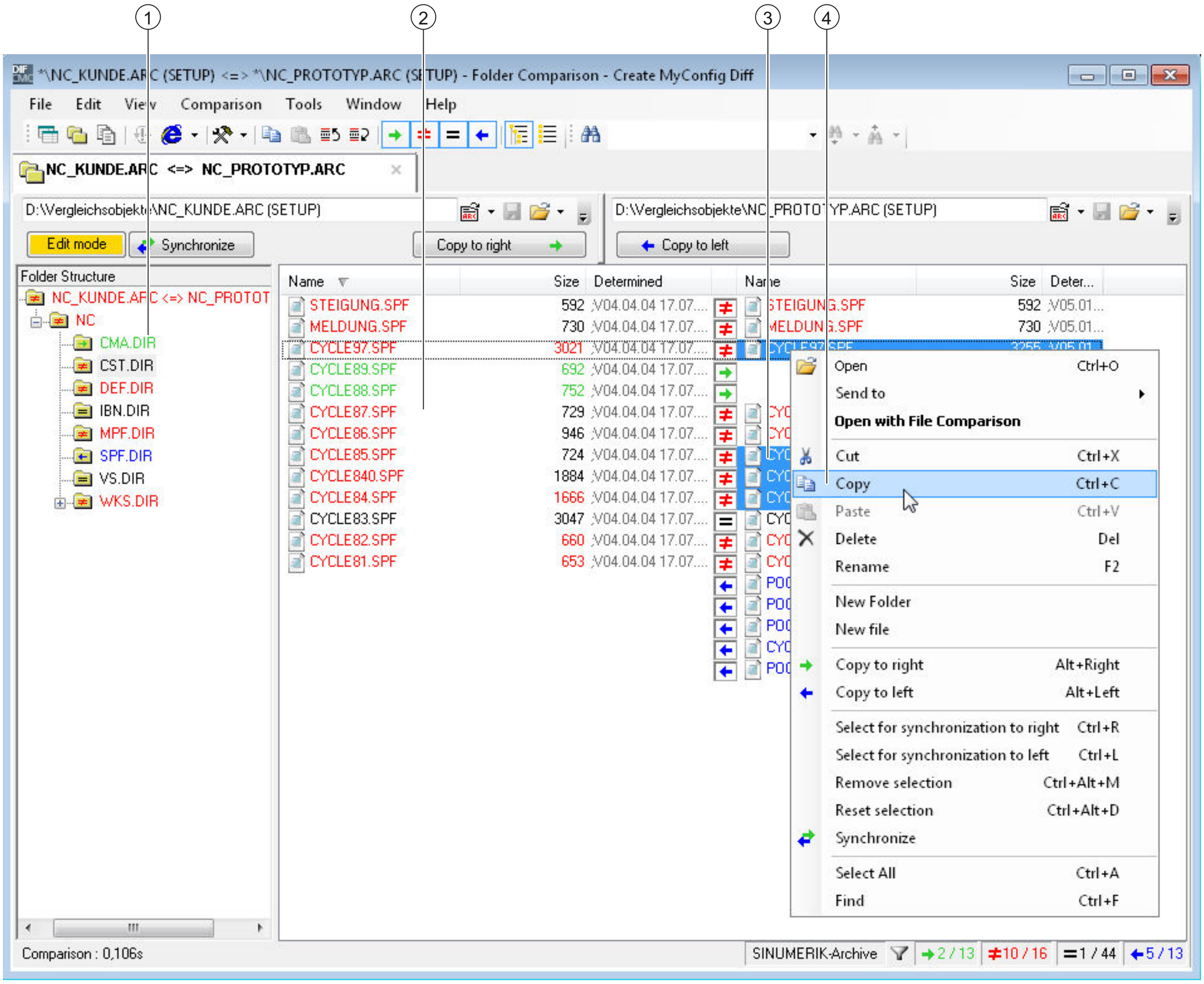
***.ini**	The "*" expression (asterisk) stands for any number of any characters. If the entry in the inclusion filter is defined, then only file names with the file extension "ini" are displayed, and all others are hidden. If "*.ini" is in the exclusion filter, then all file names with the "ini" extension are not displayed.
cycle8?.spf	The question mark represents precisely one character (any character). Files with the "cycle8" name plus any character and the "spf" file extension are displayed, and all others are hidden if the entry was defined in the inclusion filter. File names with the designation "cycle8?.spf" can therefore be called as follows: cycle80.spf, cycle81.spf, cycle8a.spf... etc.

Comment tab

Any comments on the selected comparison configuration can be saved under the Comment tab.

3.4.5 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 having to use a Create MyConfig external software.


①
②
③
④
*\NC_KUNDE.ARC (SETUP) <=> *\NC_PROTOTYP.ARC (SETUP) - Folder Comparison - Create MyConfig Diff
File
Edit
View
Comparison
Tools
Window
Help
NC_KUNDE.ARC <=> NC_PROTOTYP.ARC
D:\Vergleichsobjekte\NC_KUNDE.ARC (SETUP)
D:\Vergleichsobjekte\NC_PROTOTYP.ARC (SETUP)
Edit mode
Synchronize
Copy to right
Copy to left
Folder Structure
NC_KUNDE.ARC <=> NC_PROTOT
NC
CMA.DIR
CST.DIR
DEF.DIR
IBN.DIR
MPF.DIR
SPF.DIR
VS.DIR
WKS.DIR
Open Ctrl+O
Send to
Open with File Comparison
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Rename F2
New Folder
New file
Copy to right Alt+Right
Copy to left Alt+Left
Select for synchronization to right Ctrl+R
Select for synchronization to left Ctrl+L
Remove selection Ctrl+Alt+M
Reset selection Ctrl+Alt+D
Synchronize
Select All Ctrl+A
Find Ctrl+F
Comparison : 0,106s
SINUMERIK-Archive
2 / 13
10 / 16
1 / 44
5 / 13

Name	Size	Determined		Name	Size	Deter...
STEIGUNG.SPF	592	;V04.04.04 17.07....	≠	STEIGUNG.SPF	592	;V05.01...
MELDUNG.SPF	730	;V04.04.04 17.07....	≠	MELDUNG.SPF	730	;V05.01...
CYCLE97.SPF	3021	;V04.04.04 17.07....	≠	CYCLE97.SPF	3255	;V05.01
CYCLE89.SPF	692	;V04.04.04 17.07....	→			
CYCLE88.SPF	752	;V04.04.04 17.07....	→			
CYCLE87.SPF	729	;V04.04.04 17.07....	≠	CYC		
CYCLE86.SPF	946	;V04.04.04 17.07....	≠	CYC		
CYCLE85.SPF	724	;V04.04.04 17.07....	≠	CYC		
CYCLE840.SPF	1884	;V04.04.04 17.07....	≠	CYC		
CYCLE84.SPF	1666	;V04.04.04 17.07....	≠	CYC		
CYCLE83.SPF	3047	;V04.04.04 17.07....	=	CYC		
CYCLE82.SPF	660	;V04.04.04 17.07....	≠	CYC		
CYCLE81.SPF	653	;V04.04.04 17.07....	≠	CYC		
			←	POC		
			←	POC		
			←	POC		
			←	CYC		
			←	POC		

In the selected folder ①, in the list of folders and files contained in it ②, the objects ③ to be copied are selected using normal 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.

Objects are copied using the normal Windows tools (context menu ④, button, <Ctrl+C>, etc.).

3.5 File comparison

3.5.1 General information

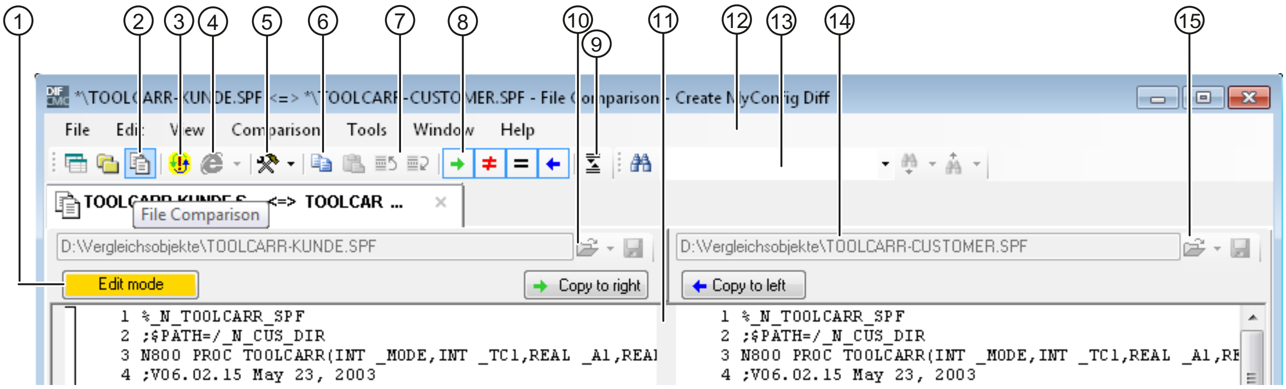
In a file comparison using Diff, you can compare the contents of two files line-by-line.

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 the 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.

3.5.2 Overview of user interface

3.5.2.1 File comparison - toolbar


①
②
③
④
⑤
⑥
⑦
⑧
⑩
⑨
⑪
⑫
⑬
⑭
⑮
*\TOOLCARR-KUNDE.SPF <=> *\TOOLCARR-CUSTOMER.SPF - File Comparison - Create MyConfig Diff
File
Edit
View
Comparison
Tools
Window
Help
TOOLCARR-KUNDE.S... <=> TOOLCAR ...
File Comparison
D:\Vergleichsobjekte\TOOLCARR-KUNDE.SPF
D:\Vergleichsobjekte\TOOLCARR-CUSTOMER.SPF
Edit mode
Copy to right
Copy to left

```
1 %_N_TOOLCARR_SPF
2 ;$PATH=/_N_CUS_DIR
3 N800 PROC TOOLCARR(INT _MODE,INT _TC1,REAL _A1,REAI
4 ;V06.02.15 May 23, 2003
```

```
1 %_N_TOOLCARR_SPF
2 ;$PATH=/_N_CUS_DIR
3 N800 PROC TOOLCARR(INT _MODE,INT _TC1,REAL _A1,RE
4 ;V06.02.15 May 23, 2003
```

① Edit mode / **Copy**

② New file comparison

Calling a file comparison.

③ Update comparison

④ Display/save HTML/XML report

⑤ Comparison configuration

⑥ Copy

⑦ Go to previous / next difference.

⑧ Group of buttons for displaying and hiding the comparison results.

⑨ Differences in context

⑩ Select file for comparison

Here selection of the **left-hand** comparison object

⑪ Column with the comparison result symbols, to separate the left-hand and right-hand files.

⑫ Switch between opened comparisons

⑬ Input field Search

⑭ Path of the opened file.

⑮ Select file for comparison

Here selection of the **right-hand** comparison object

Description of buttons/fields of the toolbar

The buttons of the toolbar are described in more detail below.

① Edit mode / Copy

The edit mode can be activated and deactivated using the "Edit mode" button. Using the "Copy to the right" and "Copy to the left" buttons, all of the selected lines are transferred from one file into the other. When the edit mode is activated, using the context menu, additional edit functions can be applied.

This is how lines can be selected, deleted, copied and reinserted. Further, text in lines can be modified. You can change the display mode using the shortcut menu.

The changes are saved in the file using the floppy disk symbol.

④ Update comparison

The update button is always shown in color if the displayed comparison result is not a current result and the comparison should be restarted using this button.

① Display/save HTML/XML report

The comparison result is shown in the standard browser (Internet Explorer) by clicking this button.
The result can be saved as HTML file or as XML protocol.

The comparison result is output completely or, under "...like display," the output depends on the settings made under ⑧.

⑥ Comparison configuration

Using this button, a comparison configuration can be changed or a different comparison configuration can be selected (see section "Configuration (Page 66)").

Note

An automatic selection of the comparison configuration as a function of the file type identification is the default setting that can be manually changed at any time.

You will find more information about this in section "Configuring automatic file identification (Page 72)."

⑥ Copy

A button to copy selected lines of the left-hand or right-hand file (see section "Copying from the data comparison (Page 74)").

⑧ Button group for showing and hiding the comparison results

The associated lines (left only, unequal, equal and right only) of the comparison result can be hidden or displayed using this group of buttons.

⑨ Differences in the context

The "Differences in the context" button hides larger blocks with identical lines.

If more than three identical lines are located before or after a different line, then these (up to three 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

⑥ Switch between opened comparisons

In the opened tabs, similar to the Window task bar, you can toggle between the opened comparisons with a mouse click or by simultaneously pressing "Ctrl" and "Tab".

→ Input field Search

Edit box for the "Search" dialog box to search for text within both files

This group contains:

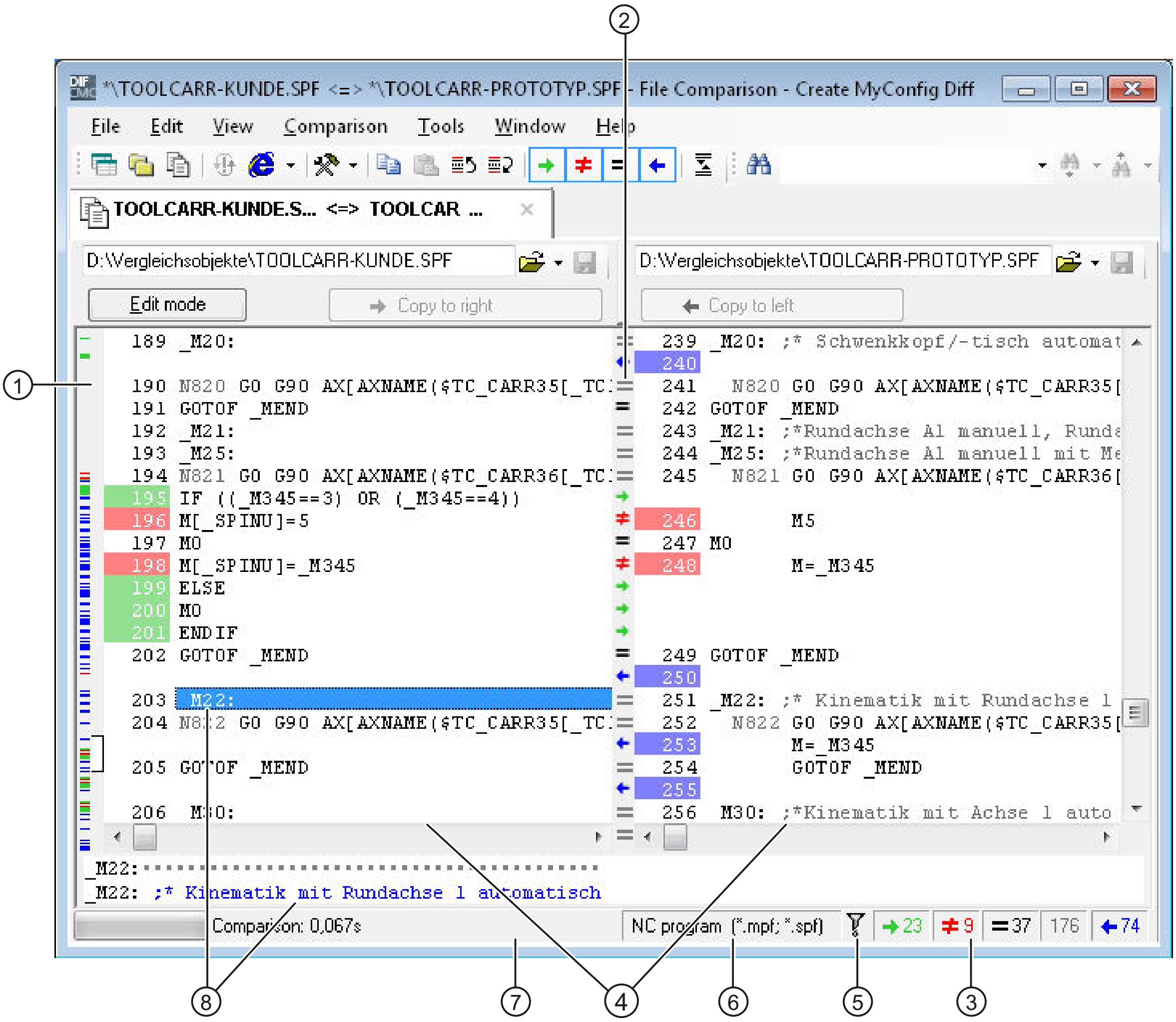
- a button for opening a dialog box in which the search options can be defined
- an edit box for searching for character strings in the display fields
- buttons for searching upward and downward.

→ Search down

→ Search up

3.5.2.2 File comparison - overview window

Below is an overview of the functions in the file comparison window.


*\TOOLCARR-KUNDE.SPF <=> *\TOOLCARR-PROTOTYP.SPF - File Comparison - Create MyConfig Diff
File
Edit
View
Comparison
Tools
Window
Help
TOOLCARR-KUNDE.S... <=> TOOLCAR ...
Edit mode
Copy to right
Copy to left
①
②
③
④
⑤
⑥
⑦
⑧

D:\Vergleichsobjekte\TOOLCARR-KUNDE.SPF

```
189 _M20:

190 N820 G0 G90 AX[AXNAME($TC_CARR35[_TC:
191 GOTOF _MEND
192 _M21:
193 _M25:
194 N821 G0 G90 AX[AXNAME($TC_CARR36[_TC:
195 IF ((_M345==3) OR (_M345==4))
196 M[_SPINU]=5
197 M0
198 M[_SPINU]=_M345
199 ELSE
200 M0
201 ENDIF
202 GOTOF _MEND

203  M22:
204 N8:2 G0 G90 AX[AXNAME($TC_CARR35[_TC:

205 GO'OF _MEND

206  M:0:
```

D:\Vergleichsobjekte\TOOLCARR-PROTOTYP.SPF

```
239 _M20: ;* Schwenkkopf/-tisch automat
240
241   N820 G0 G90 AX[AXNAME($TC_CARR35[
242 GOTOF _MEND
243 _M21: ;*Rundachse A1 manuell, Runda
244 _M25: ;*Rundachse A1 manuell mit Me
245   N821 G0 G90 AX[AXNAME($TC_CARR36[

246         M5
247 M0
248         M=_M345



249 GOTOF _MEND
250
251 _M22: ;* Kinematik mit Rundachse 1
252   N822 G0 G90 AX[AXNAME($TC_CARR35[
253         M=_M345
254         GOTOF _MEND
255
256  M30: ;*Kinematik mit Achse 1 auto
```

```
_M22:·······································
_M22: ;* Kinematik mit Rundachse 1 automatisch
```

Comparison: 0,067s | NC program (*.mpf; *.spf) | ➜23 | ≠9 | =37 | 176 | ←74

① Overview comparison result

② Column with the comparison result symbols, to separate the left-hand and right-hand files.

③ ③ Specified values for lines (status bar bottom right)

④ Comparison result symbols 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 Data comparison window

The individual functionalities of the file comparison are described in more detail below:

① Overview comparison result

The comparison result is shown over the complete content of both files in this bar as an overview.

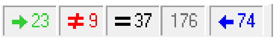
The colored dashes represent lines that are only available on the left-hand side, only on the right-hand side – or are not equal.

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 can be positioned by clicking on the toolbar.

② Column with comparison result symbols

See description for ④

③ Specified values for lines (status bar bottom right)



→23 ≠9 =37 176 ←74

These values provide information about how many lines are either available only in the left-hand file (green), where the contents differ (red), are identical in both files (black), are identical in the relevant part of the line (gray) or only exist in the right-hand file (blue).

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 unequal character and the line numbers with red background indicate that the content of the line is not identical.

= The black equal character indicates that both lines are identical.

= The gray equal character indicates matches in the relevant part of the line.

Note

Non-relevant parts of a line (e.g. spaces or comments) can be defined in the comparison configuration (see section Configuration).

← The blue arrow and the line number with blue background indicate that this line only exists in the right-hand file.

Note

Lines or characters shown in gray have been excluded from a comparison by a filter.

---

⑥ Comparison configuration - Filter

A filter is active in the comparison configuration. The comparison configuration is opened with a double-click.

Note

Character strings that are not relevant can be excluded from the comparison using the inclusion/exclusion filter.
In the comparison result, these character strings are shown in gray.

① Display of the current comparison configuration

The comparison configuration that is being used in the actual comparison is displayed in this field.
The comparison configuration is opened with a double-click.

⑧ Display of the lines to be compared one on top of the other

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.

With this representation it can be seen whether the characters in the lines are unequal (red), equal (black), only available at the left (green) or only available at the right (blue).

A gray point • represents a character that does not exist in this line. In this case, there is a character at this location in the other line (possibly also a space).

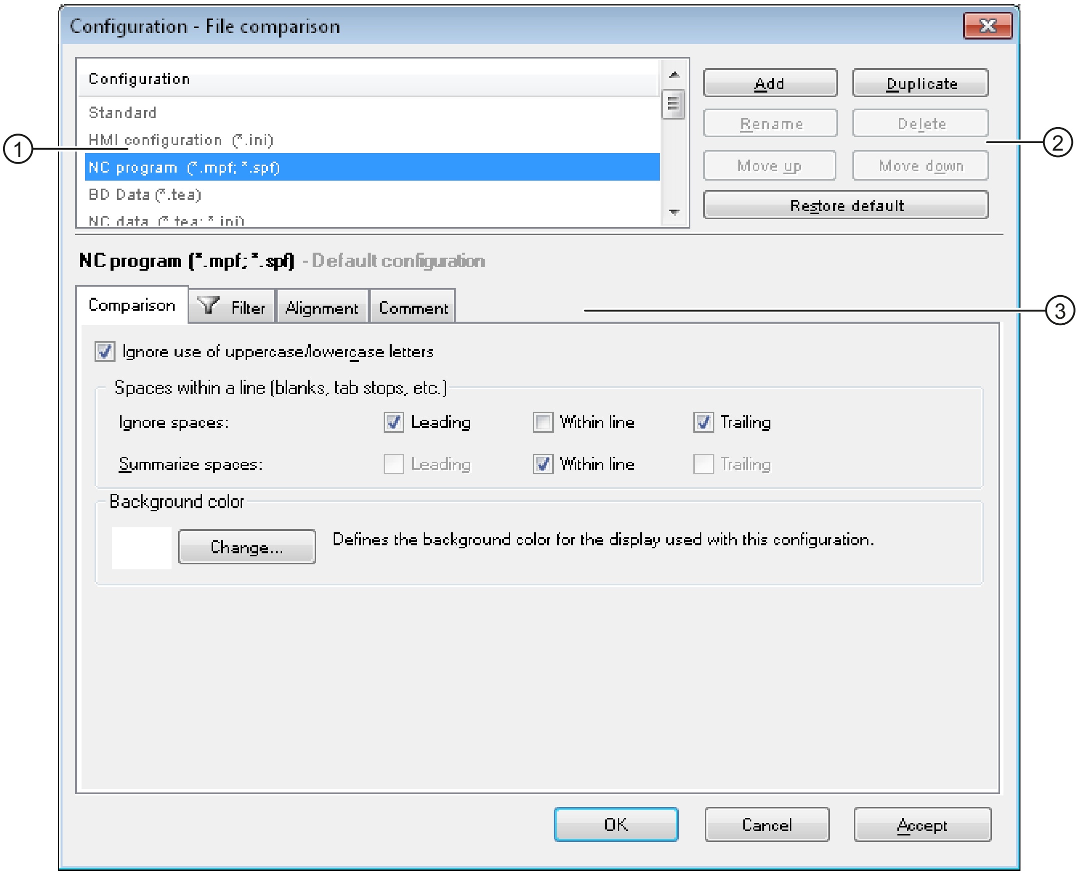
See also

Configuration (Page 66)

3.5.3 Configuration

File comparison

You can call the dialog to configure a folder comparison using the "Comparison Configuration" 🛠 button or via the menu "Comparison > Edit Configuration…". You can manage the comparison configurations (i.e. create, edit, delete, etc.) in the dialog.
In a comparison configuration, all of the criteria that should be taken into account when comparing files can be defined.
You can also export or import configurations, for example, to make them available to other users (see Exporting and importing configurations (Page 75)).



Configuration
Standard
HMI configuration (*.ini)
NC program (*.mpf; *.spf)
BD Data (*.tea)
NC data (*.tea; *.ini)

NC program (*.mpf; *.spf) - Default configuration

Comparison | Filter | Alignment | Comment

☑ Ignore use of uppercase/lowercase letters

Spaces within a line (blanks, tab stops, etc.)

Ignore spaces:	☑ Leading	☐ Within line	☑ Trailing
Summarize spaces:	☐ Leading	☑ Within line	☐ Trailing

Background color

Change... Defines the background color for the display used with this configuration.

① The existing comparison configurations are listed in this area of the configuration dialog and can be selected by clicking on them. Configurations shown in gray are user-specified configurations; these can be changed, but not deleted - and can be restored again at any time.

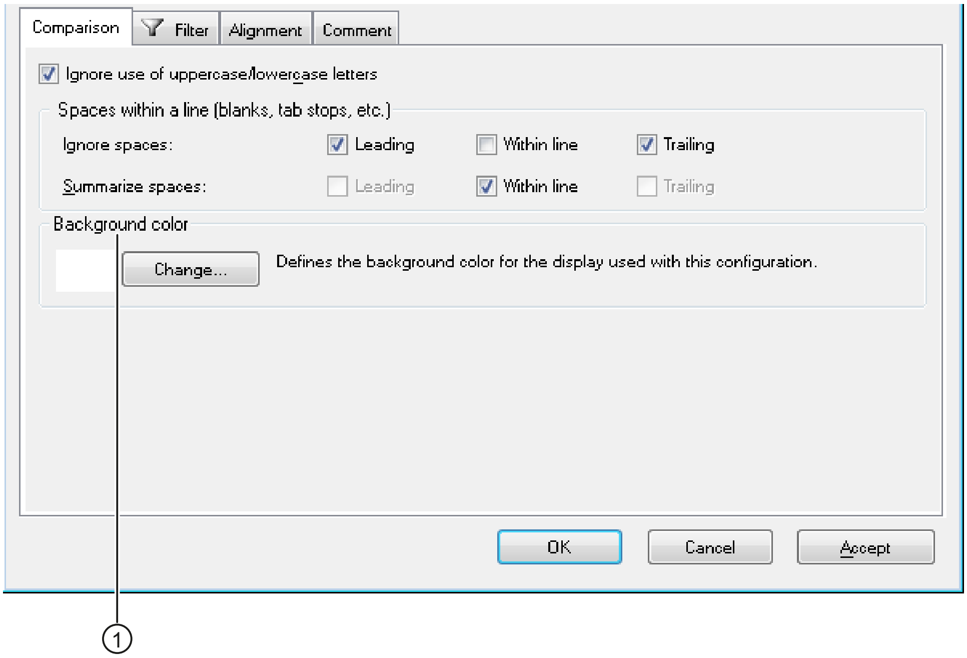
② Configurations can be managed using these buttons.
Using "Add," a new configuration is generated, which can also be adapted.
A copy of the selected comparison configuration is created by clicking "Duplicate."
"Restore user-specified setting" undoes the changes to the specified comparison configuration.

NOTE::
The "Restore user-specified setting" 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 the "Compare" tab. The background color for comparisons can additionally be defined.



☑ Ignore use of uppercase/lowercase letters

Spaces within a line (blanks, tab stops, etc.)

Ignore spaces: ☑ Leading □ Within line ☑ Trailing

Summarize spaces: □ Leading ☑ Within line □ Trailing

Background color

Defines the background color for the display used with this configuration.

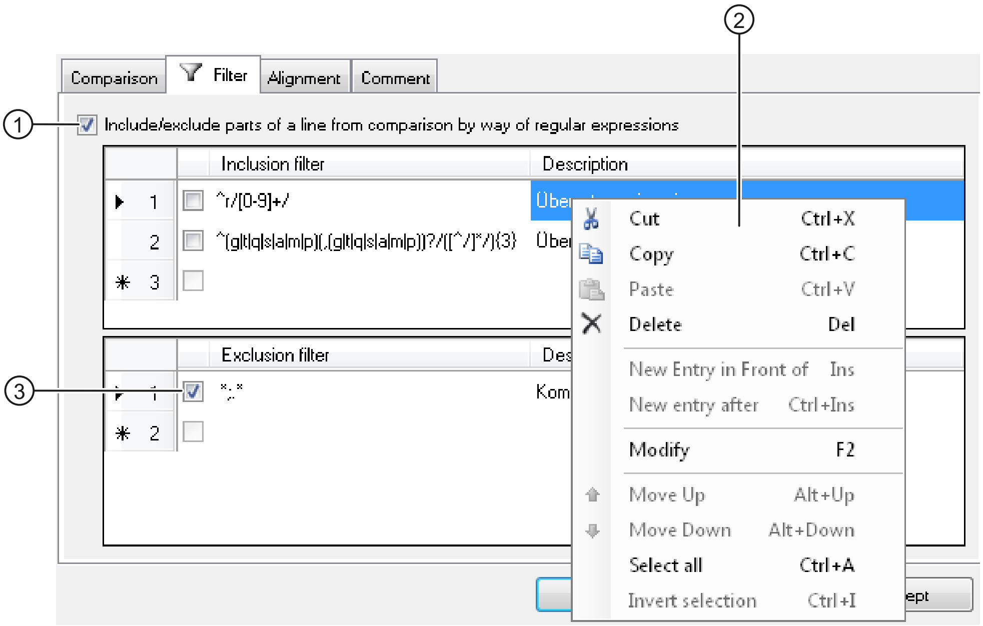
① **Background color**

So that the user can immediately recognize how to select certain comparison configurations, a separate background color can be defined for each comparison configuration.

Filter tab

Using filters, parts of a line can be included in or excluded from the comparison based on regular expressions.
Characters excluded from the comparison are shown in gray in the comparison results. This means that the contents of a file are always displayed completely, but marked in color.

If only the relevant characters are identical in the line but non-relevant characters are different, the comparison symbol will be a gray equals sign.



☑ Include/exclude parts of a line from comparison by way of regular expressions

		Inclusion filter	Description
▶	1	□ ^r/[0-9]+/	Über
	2	□ ^(g|t|q|s|a|m|p)(,(g|t|q|s|a|m|p))?/([^/]*/){3}	Über
✱	3	□	

		Exclusion filter	Des
▶	1	☑ *;.*	Kom
✱	2	□	

① The functionality of the inclusion and exclusion filter is activated using a check mark for "Parts of a line..."

NOTE:
The filter symbol is shown in the status bar if this function is activated.

② Filter entries can be created, edited and deleted via the context menu.

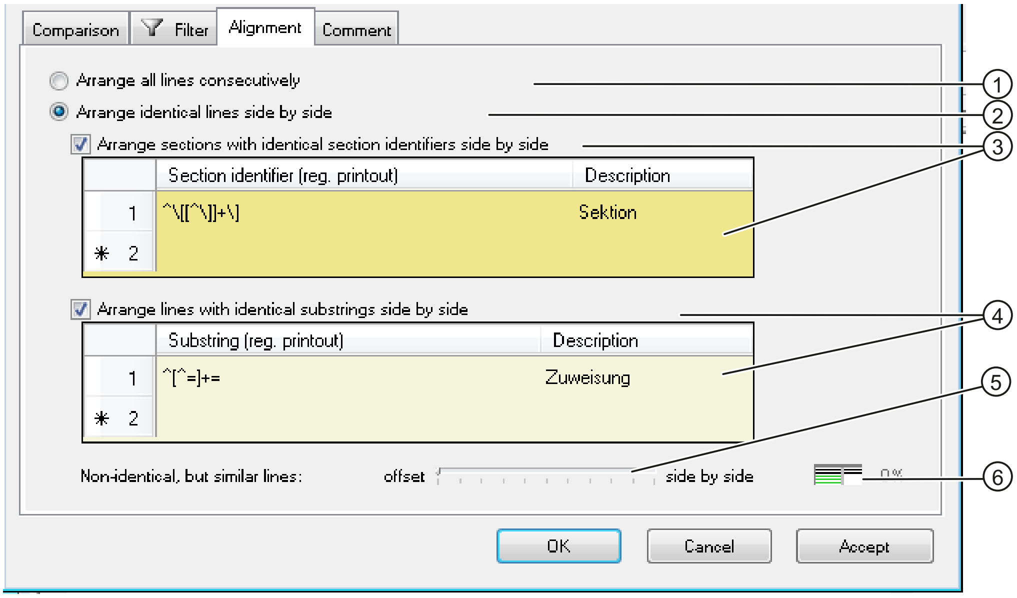
③ Every filter entry can be individually activated or deactivated.

Examples of filters:

;.	The regular expression (see Use of regular expressions in Diff (Page 79)) 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 leading characters as prefix 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.

Alignment tab

Under the "Alignment" tab, you can define how the individual lines of both files should be aligned with one another after the comparison.



○ Arrange all lines consecutively

● Arrange identical lines side by side

☑ Arrange sections with identical section identifiers side by side

	Section identifier (reg. printout)	Description
1	^\[[^\]]+\]	Sektion
* 2		

☑ Arrange lines with identical substrings side by side

	Substring (reg. printout)	Description
1	^[^=]+=	Zuweisung
* 2		

Non-identical, but similar lines: offset side by side 0 %

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, etc.).
A search is not made across lines regarding identity or similarity.

② As an alternative to ①, in practice, this selection is more frequently used, which defines that identical lines should be aligned next to one another.
This selection can be further refined or extended by making the appropriate entries for 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 context menu or double-click) a regular expression (see Use of regular expressions in Diff (Page 79)) should be saved in the entry field to describe the section identifier.

NOTE:
The 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 to describe the partial character string must be saved in the entry field.
For example, this function is used to define that in files which have the structure "identifier= value", only identical identifiers are placed next to one another (compared).

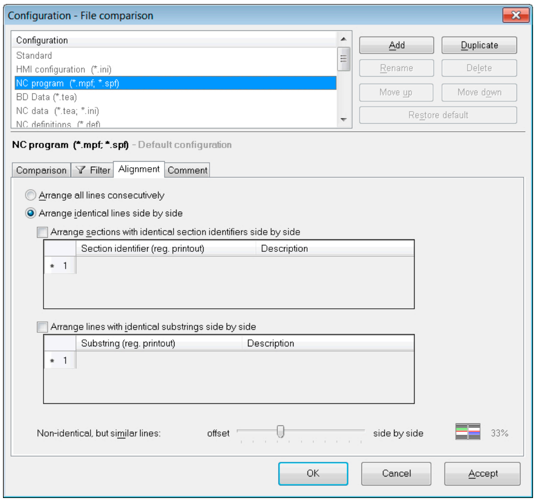
NOTE:
The lines that are found also have a light yellow background in the comparison display.

NOTICE:
When selection ③ is activated, lines with identical partial character strings are only placed next to one another (compared) if they are located in the same section.

⑤ In addition, with lines for which the definitions ③ and ④ do not apply, this slider can be used to set 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 symbol).

⑥ This diagram should symbolically explain how the slider functions.

Two typical comparison configurations for SINUMERIK are subsequently shown and their comparison results as example.



Configuration - File comparison

Configuration: Standard; HMI configuration (*.ini); **NC program (*.mpf; *.spf)**; BD Data (*.tea); NC data (*.tea; *.ini); NC definitions (*.def)

NC program (*.mpf; *.spf) - Default configuration

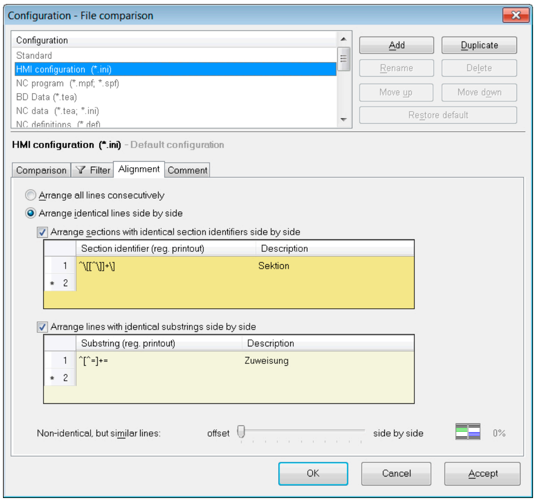
Alignment tab:
- ○ Arrange all lines consecutively
- ● Arrange identical lines side by side
- □ Arrange sections with identical section identifiers side by side

	Section identifier (reg. printout)	Description
1		

- □ Arrange lines with identical substrings side by side

	Substring (reg. printout)	Description
1		

Non-identical, but similar lines: offset — side by side 33%



Configuration - File comparison

Configuration: Standard; **HMI configuration (*.ini)**; NC program (*.mpf; *.spf); BD Data (*.tea); NC data (*.tea; *.ini); NC definitions (*.def)

HMI configuration (*.ini) - Default configuration

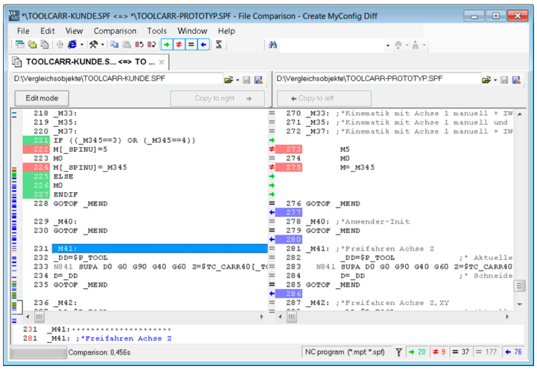
Alignment tab:
- ○ Arrange all lines consecutively
- ● Arrange identical lines side by side
- ☑ Arrange sections with identical section identifiers side by side

	Section identifier (reg. printout)	Description
1	^\[[^\]]+\]	Sektion
2		

- ☑ Arrange lines with identical substrings side by side

	Substring (reg. printout)	Description
1	^[^=]+=	Zuweisung
2		

Non-identical, but similar lines: offset — side by side 0%

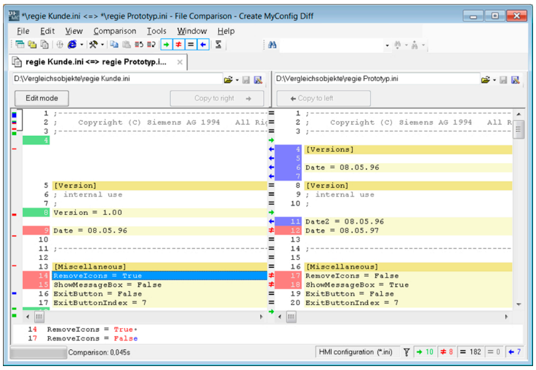


*\TOOLCARR-KUNDE.SPF <=> *\TOOLCARR-PROTOTYP.SPF - File Comparison - Create MyConfig Diff

D:\Vergleichsobjekte\TOOLCARR-KUNDE.SPF | D:\Vergleichsobjekte\TOOLCARR-PROTOTYP.SPF

```
218 _M33:                                  = 270 _M33: ;*Kinematik mit Achse 1 manuell + IW
219 _M35:                                  = 271 _M35: ;*Kinematik mit Achse 1 manuell und
220 _M37:                                  = 272 _M37: ;*Kinematik mit Achse 1 manuell + IW
221 IF ((_M345==3) OR (_M345==4))          →
222 M[_SPINU]=5                            ≠ 273         M5
223 M0                                     = 274         M0
224 M[_SPINU]=_M345                        ≠ 275         M=_M345
225 ELSE                                   →
226 M0                                     →
227 ENDIF                                  →
228 GOTOF _MEND                            = 276 GOTOF _MEND
                                           ← 277
229 _M40:                                  = 278 _M40: ;*Anwender-Init
230 GOTOF _MEND                            = 279 GOTOF _MEND
                                           ← 280
231 _M41:                                  = 281 _M41: ;*Freifahren Achse Z
232 _DD=$P_TOOL                            = 282        _DD=$P_TOOL              ;* Aktuelle
233 N841 SUPA D0 G0 G90 G40 G60 Z=$TC_CARR40[_TC= 283 N841 SUPA D0 G0 G90 G40 G60 Z=$TC_CARR40
234 D=_DD                                  = 284        D=_DD                    ;* Schneide
235 GOTOF _MEND                            = 285 GOTOF _MEND
                                           ← 286
236 _M42:                                  = 287 _M42: ;*Freifahren Achse Z,XY

231  _M41:·····················
281  _M41: ;*Freifahren Achse Z
```

Comparison: 0,456s | NC program (*.mpf; *.spf) | → 20 | ≠ 9 | = 37 | = 177 | ← 76



*\regie Kunde.ini <=> *\regie Prototyp.ini - File Comparison - Create MyConfig Diff

D:\Vergleichsobjekte\regie Kunde.ini | D:\Vergleichsobjekte\regie Prototyp.ini

```
 1 ;------------------------------------  =  1 ;------------------------------------
 2 ;    Copyright (C) Siemens AG 1994    All Ri=  2 ;    Copyright (C) Siemens AG 1994    All R
 3 ;------------------------------------  =  3 ;------------------------------------
 4                                        →
                                          ←  4 [Versions]
                                          ←  5
                                          ←  6 Date = 08.05.96
                                          ←  7
 5 [Version]                              =  8 [Version]
 6 ; internal use                         =  9 ; internal use
 7 ;                                      = 10 ;
 8 Version = 1.00                         →
                                          ← 11 Date2 = 08.05.96
 9 Date = 08.05.96                        ≠ 12 Date = 08.05.97
10                                        = 13
11 ;------------------------------------  = 14 ;------------------------------------
12                                        = 15
13 [Miscellaneous]                        = 16 [Miscellaneous]
14 RemoveIcons = True                     ≠ 17 RemoveIcons = False
15 ShowMessageBox = False                 ≠ 18 ShowMessageBox = True
16 ExitButton = False                     = 19 ExitButton = False
17 ExitButtonIndex = 7                    = 20 ExitButtonIndex = 7

14  RemoveIcons = True·
17  RemoveIcons = False
```

Comparison: 0,045s | HMI configuration (*.ini) | → 10 | ≠ 8 | = 182 | = 0 | ← 7

Comment tab

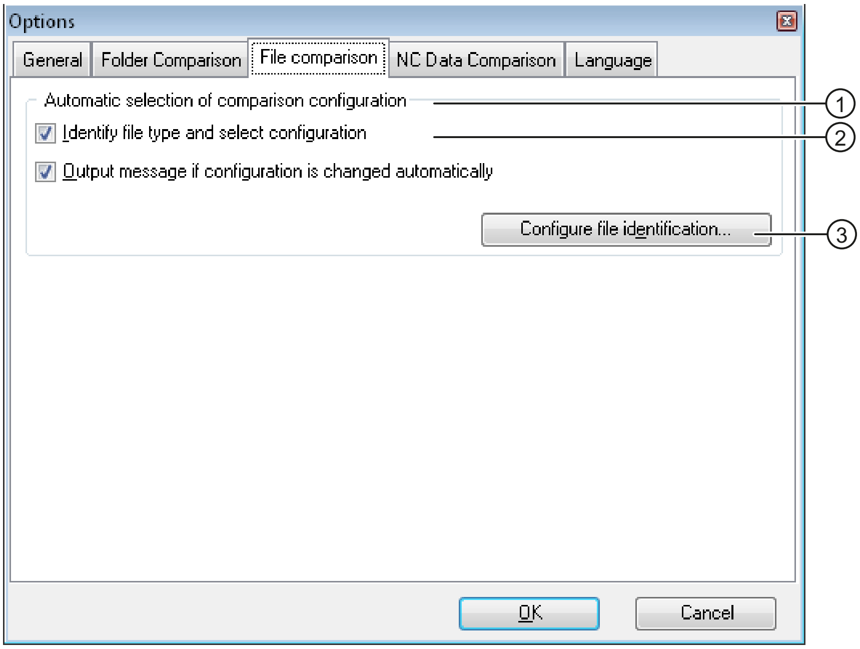
Any comments on the selected comparison configuration can saved under the Comment tab.

3.5.4 Configuration of the automatic file identification

Automatic file identification in Diff

An automatic file identification function is integrated in Diff for file comparison. This file identification function can identify the file type from the file name or the file content and can select an associated comparison configuration.

You can call the dialog for setting the program options via the menu "Tools > Options".



Options

General | Folder Comparison | File comparison | NC Data Comparison | Language

Automatic selection of comparison configuration

☑ Identify file type and select configuration

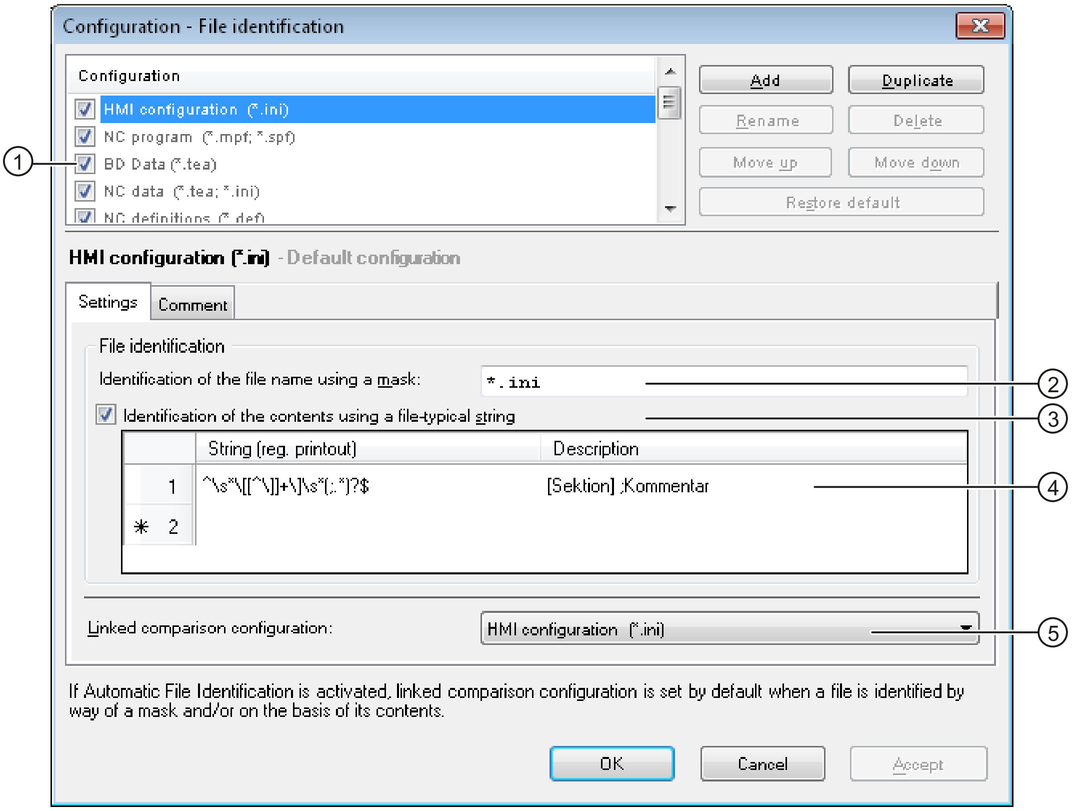
☑ Output message if configuration is changed automatically

① The automatic selection of the comparison configuration can be activated or deactivated under the "File comparison" tab.

② Further, it can be set as to whether a message should be issued when an automatic comparison configuration is changed as a result of the file identification.

③ This button can be used to call the dialog to configure the automatic file identification function.

The dialog to configure the automatic file identification function is equivalent to the dialog for the comparison configuration and should be used in the same way.



Configuration

- ☑ HMI configuration (*.ini)
- ☑ NC program (*.mpf; *.spf)
- ☑ BD Data (*.tea)
- ☑ NC data (*.tea; *.ini)
- ☑ NC definitions (*.def)

HMI configuration (*.ini) - Default configuration

File identification

Identification of the file name using a mask: *.ini

☑ Identification of the contents using a file-typical string

	String (reg. printout)	Description
1	^\s*\[[^\]]+\]\s*(;.*)?$	[Sektion] ;Kommentar
* 2		

Linked comparison configuration: HMI configuration (*.ini)

If Automatic File Identification is activated, linked comparison configuration is set by default when a file is identified by way of a mask and/or on the basis of its contents.

① When a file is opened, Diff goes through the saved identification configurations from the top to the bottom - and applies the first appropriate configuration corresponding to what has been configured under the Settings tab.
As a consequence, the sequence in which the configurations were saved is of significance.

For instance, a configuration that only operates with the file identification function using the file names should be located after a configuration that operates with the file identification function using the same file names and in addition, the file content.
Configurations that are to be temporarily not used can be deactivated by removing the checkmark.

Settings tab

② A screen form to identify the file (file type) using the file names can be saved in this field.

③ If, using the file name, a clear identification is not possible, then a character string in the content of the file - typical for the file type - can be used to identify it.

④ In this field, when activating the content identification via ③, a regular expression that describes a character string typical for the particular file type must be saved. Several regular expressions (one for each line) can also be specified. The comparison configuration is used as soon as one matches, i.e. a defined character string was found in the file.

⑤ This list box is used to specify which comparison configuration will be used if a file described using ②, ③, and ④ is identified. The drop down list box contains all of the file comparison configurations (see Configuration (Page 66)).

Note

For the user-specified configurations, the file identification configuration and the associated file comparison configuration have the same names. It is not mandatory that the names are the same. The configuration names have no significance from a functional perspective.

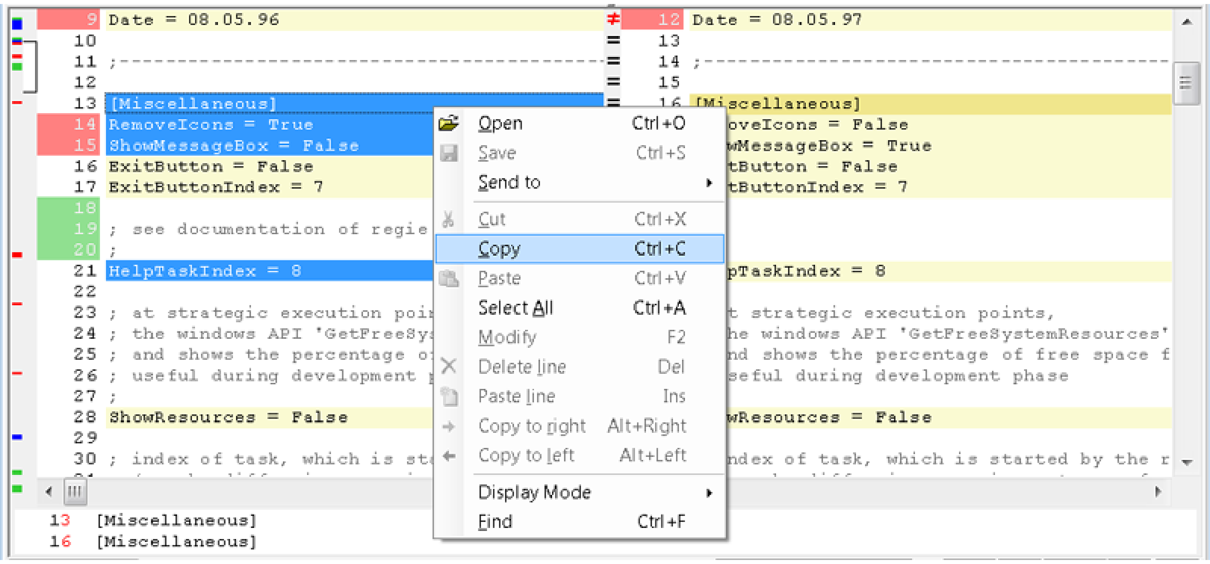
Comment tab

Any comments on the selected identification configuration can be saved under the Comment tab.

3.5.5 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.



```
 9 Date = 08.05.96
10
11 ;----------------------------------------
12
13 [Miscellaneous]
14 RemoveIcons = True
15 ShowMessageBox = False
16 ExitButton = False
17 ExitButtonIndex = 7
18
19 ; see documentation of regie
20 ;
21 HelpTaskIndex = 8
22
23 ; at strategic execution poi
24 ; the windows API 'GetFreeSy
25 ; and shows the percentage o
26 ; useful during development
27 ;
28 ShowResources = False
29
30 ; index of task, which is st
```

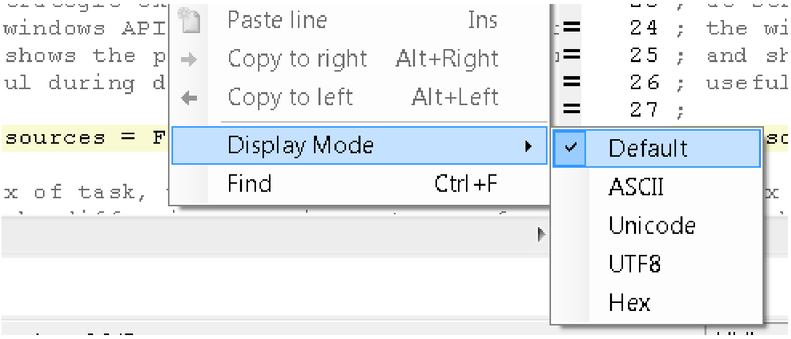
```
12 Date = 08.05.97
13
14 ;----------------------------------------
15
16 [Miscellaneous]
oveIcons = False
wMessageBox = True
tButton = False
tButtonIndex = 7


pTaskIndex = 8

t strategic execution points,
he windows API 'GetFreeSystemResources'
nd shows the percentage of free space f
seful during development phase

wResources = False

ndex of task, which is started by the r
```

```
13   [Miscellaneous]
16   [Miscellaneous]
```

Objects are copied using the normal Windows tools (context menu, button, <Ctrl+C>, etc.).

3.5.6 Display modes

The display mode for the file to be compared can be selected using the context menu of the file comparison.



3.6 Exporting and importing configurations

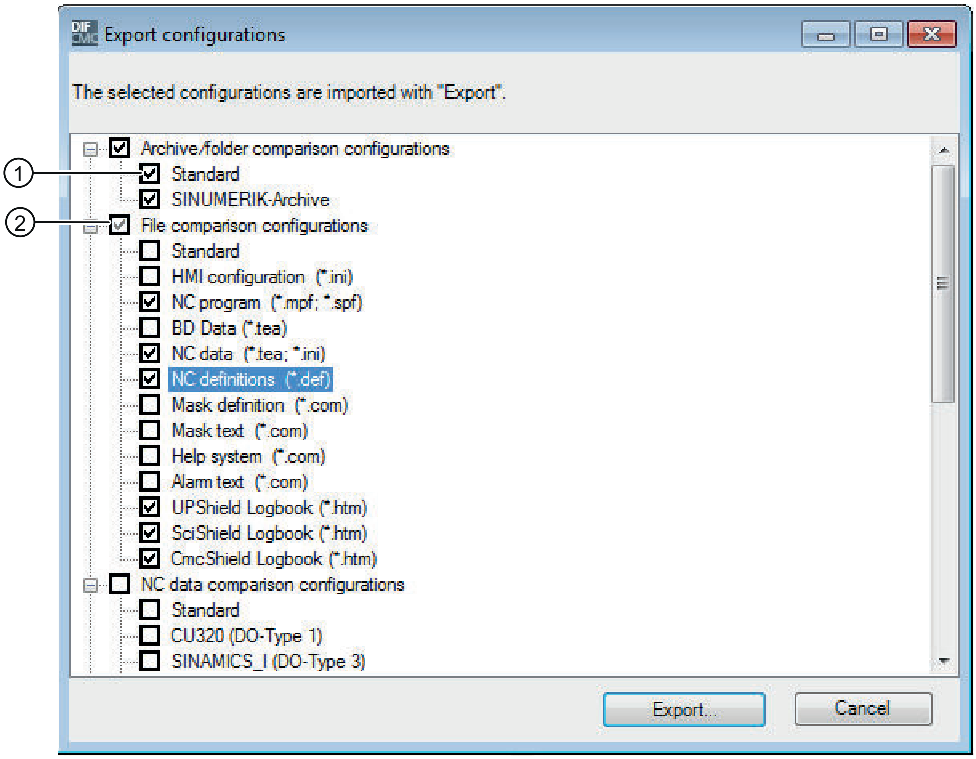
3.6.1 General information

The comparison configurations, specified in Diff and those that you configured yourself, file identification configurations and data task configurations can be exported in a file, either individually or in groups, and imported 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.

3.6.2 Exporting

The dialog to export configurations can be called via the main menu "Tools > Export Configurations".



Export configurations

The selected configurations are imported with "Export".

- ☑ Archive/folder comparison configurations
  - ☑ Standard
  - ☑ SINUMERIK-Archive
- ☑ File comparison configurations
  - □ Standard
  - □ 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 text (*.com)
  - ☑ UPShield Logbook (*.htm)
  - ☑ SciShield Logbook (*.htm)
  - ☑ CmcShield Logbook (*.htm)
- □ NC data comparison configurations
  - □ Standard
  - □ CU320 (DO-Type 1)
  - □ SINAMICS_I (DO-Type 3)

Export... | Cancel

The configurations available in Diff are displayed and grouped according to application areas.

① By activating the individual configurations, for each configuration you can define whether this should be exported.

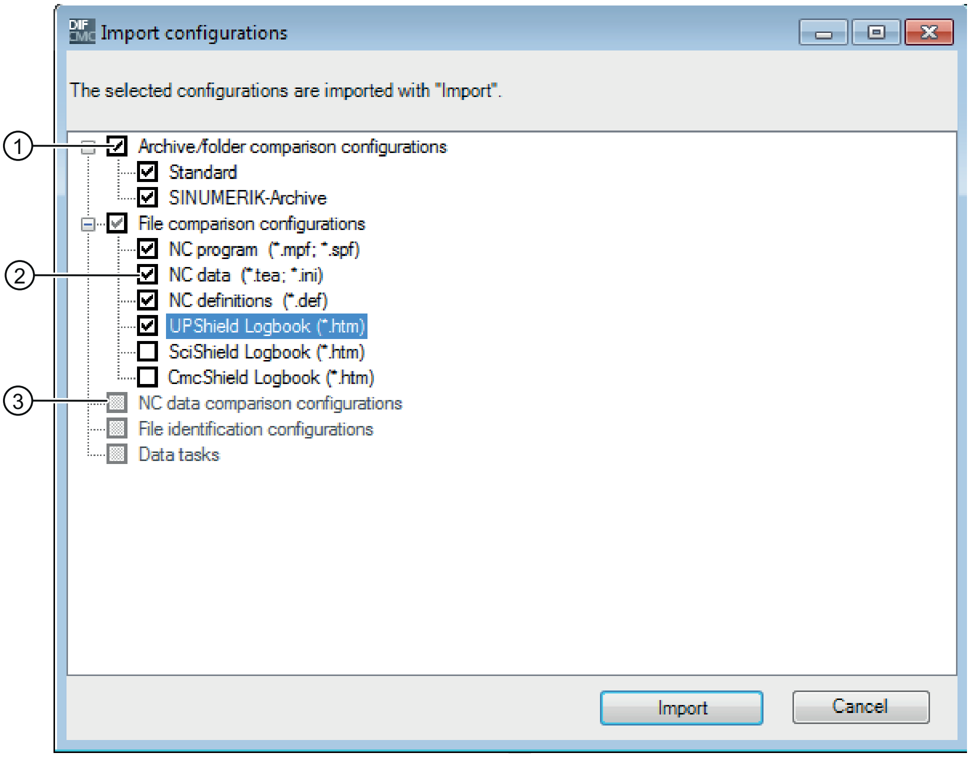
② Activating/deactivating a group node means that this action is accepted for all configurations of the group.

The "Export" button opens the "Save as" dialog which saves the configurations to a file.

3.6.3 Import

Configurations can be imported 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.


Import configurations
The selected configurations are imported with "Import".
☑ Archive/folder comparison configurations
☑ Standard
☑ SINUMERIK-Archive
☑ File comparison configurations
☑ NC program (*.mpf; *.spf)
☑ NC data (*.tea; *.ini)
☑ NC definitions (*.def)
☑ UPShield Logbook (*.htm)
□ SciShield Logbook (*.htm)
□ CmcShield Logbook (*.htm)
▨ NC data comparison configurations
▨ File identification configurations
▨ Data tasks
Import
Cancel
①
②
③

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 prompted as to whether the existing configuration should be overwritten.

3.7 Command line call

To be able to use Diff via the shortcut menu, e.g. in Windows Explorer, Diff has been provided with option switches which, in turn, allow Diff to be called via the command line.

Call

```
Diff /data <archive path 1> [< archive path 2>] [< archive path n>]
Diff /dir <directory path 1> [< directory path 2>]
Diff /file <file path 1> [< file path 2>]
```

Parameter

Table 3-2 Parameter

Options	Arguments	Description
/data	<"Archive(s)">	• Means that the NC data comparison is opened • As argument(s), one or more paths for archives or parameter files can be specified here, which are to be compared • The path must be specified in double inverted commas if spaces are included in it (example: "C:\path with spaces\")
/dir	<"Directory(ies)">	• Means that the folder comparison is opened • As argument(s), one or two paths can be specified here for directories or archives that are to be compared • The path must be specified in double inverted commas if spaces are included in it (example: "C:\path with spaces\")
/file	<"File(s)">	• Means that the file comparison is opened • As argument(s), one or two paths can be specified here for files that are to be compared • The path must be specified in double inverted commas if spaces are included in it (example: "C:\path with spaces\")
/config	<"Comparison configuration">	A name of a comparison configuration can be specified. If the configuration exists, then this is used. **NOTICE** Case sensitive.

Return values

Table 3-3 Return values

-	No evaluation possible

Example

(Requirement: Create MyConfig is installed in the directory "C:\Program Files\SIEMENS \Create MyConfig 4.7" an archive "Archive1.arc" is available in the path "C:\Temp\" and in Diff there is the comparison configuration "myConfig".)

```
"C:\Program Files\SIEMENS\Create MyConfig 4.7\Diff.exe" /data "C:
\Temp\Archive1.arc" /config "myConfig"
```

A Use of regular expressions in Diff

A.1 General information on regular expressions

Regular expressions represent a general notation to describe text patterns. They are frequently 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 if required to filter it (e.g. all block numbers or all comments in a cycle).

A regular expression comprises two types of characters, literal text characters and metacharacters:

Literal characters

Literal characters are characters that only stand for themselves in a regular expression and have no special significance.
A search with a regular expression which only comprises literals, therefore corresponds to a conventional character search.

Meta characters

Meta characters are characters that have a special function within regular expressions. Meta characters are, e.g. the characters $ ^ { } [ ] ( ) | . * + ? \ -.

Analogous to natural languages, literals can be considered as vocabulary and the meta characters as 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 realized by using a backslash \ as prefix.
For example, if an open square bracket [is to be identified with a regular expression, then it must have a backslash \ as prefix; the reason for this is that the square bracket alone is a meta character, i.e. has a special function (see character classes). This means, in this case that \
[must be written in the regular expression.

A.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.
`[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 a 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 (logical 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)_`

Meta characters	Description
`^`	The meta character for the beginning of the line does not describe a character in the actual sense, but instead a position in a line. Example: `^[aA]` requires a lower-case `a` or 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.
`$`	The meta character for the end of the line does not describe a character in the actual sense, but instead a position in a line. Example: `[aA]$` 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}`.

A.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.

Regular expression	Charac-ter	Description
`^\[[^\]]+\]`	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
`^\[[^\]]+\]`	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	Charac-ter	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**`\(\d+\)`	② `CHANDATA`	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.
`^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 4

4.1 Safety instructions

4.1.1 Fundamental safety instructions

4.1.1.1 General safety instructions

⚠ WARNING

Risk of death if the safety instructions and remaining risks are not carefully observed

If the safety instructions and residual risks are not observed in the associated hardware documentation, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

⚠ WARNING

Danger to life or malfunctions of the machine as a result of incorrect or changed parameterization

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

- Protect the parameterization (parameter assignments) against unauthorized access.
- Respond to possible malfunctions by applying suitable measures (e.g. EMERGENCY STOP or EMERGENCY OFF).

4.1.1.2 Industrial security

Note

Industrial security

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

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

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

⚠ **WARNING**

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

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

- Keep the software up to date.
  You will find relevant information and newsletters at this address (http://support.automation.siemens.com).
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
  You will find further information at this address (http://www.siemens.com/industrialsecurity).
- Make sure that you include all installed products into the holistic industrial security concept.

4.2 Introduction

Note

Setup of documentation: Operating Manual / online help

The contents for the **Expert** section are identical in the Operating Manual and online help.

Expert is the engineering software for creating a CMC project. An executable package for the production and upgrading of machines can be generated from this project with the "Deployment" function. The package contains a configurable sequence of production or upgrading steps and the associated data.

In Expert, the configuration of the package, the dialog boxes for operator interactions, and the individual upgrade steps in the form of a hierarchical step tree are defined and the data and files that are required or will be manipulated are specified.
The configuration can only be performed by qualified personnel.

The majority of the settings can be easily made via a context-sensitive context menu.

A source explorer (similar to the Windows Explorer) is integrated for the integration of files in the components. Files can be either copied or linked. Linked files are only accepted in the package at the time of transfer.

Before a Verification of package, you will be informed if there are any errors in the configuration.

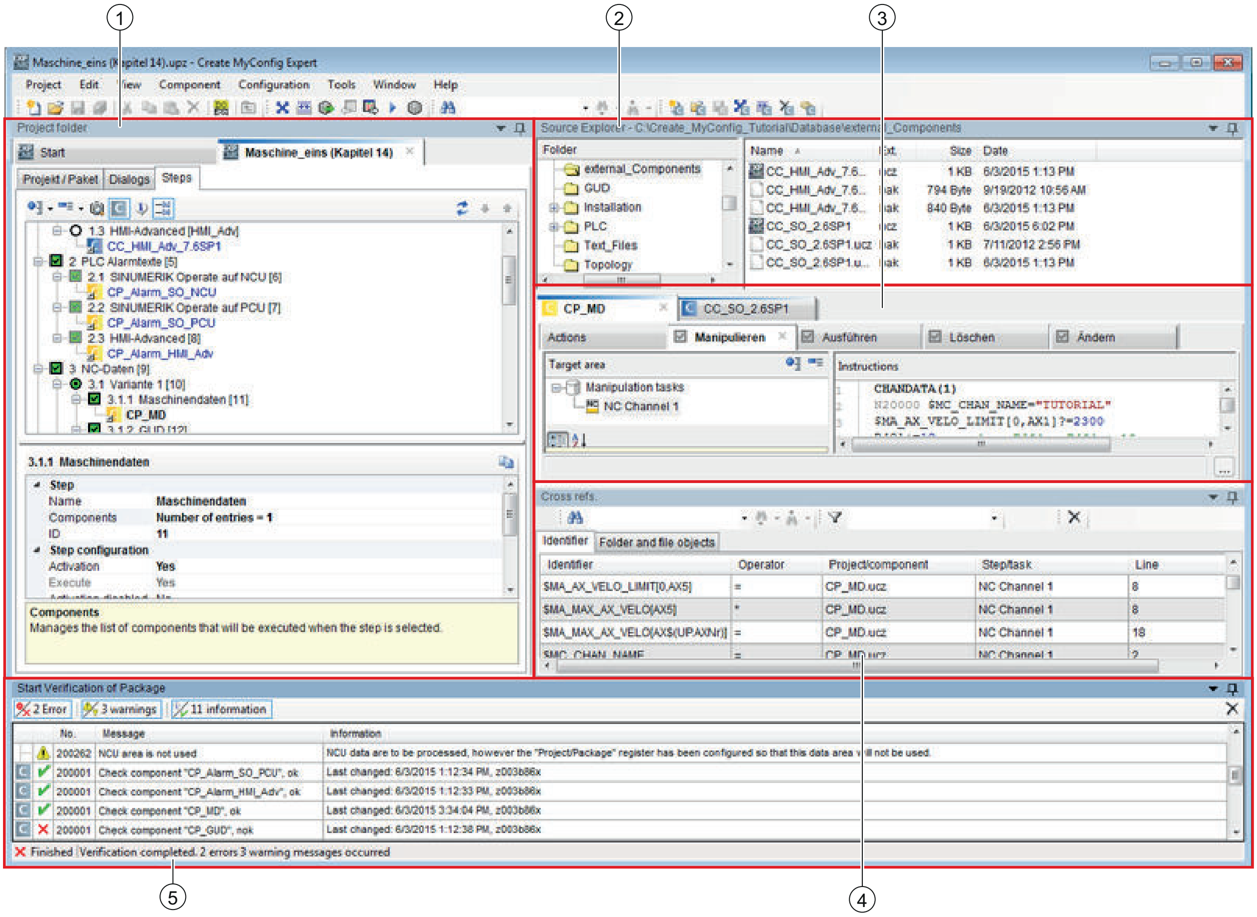
The "Transfer" function creates a package that can be executed under Linux or Windows depending on the configuration.

An integrated online help (F1), which is identical to this documentation, supports you in the correct configuration.

4.2.1 Overview of the expert user interface

The expert user interface comprises the following areas:

- Main menus and toolbars (Page 88)
- Project folder (Page 92)
- Source explorer (Page 91)
- Component editor (Page 183)
- Cross references (Page 236)
- Verification of package



3.1.1 Maschinendaten

Step	
Name	Maschinendaten
Components	Number of entries = 1
ID	11
Step configuration	
Activation	Yes
Execute	Yes
Activation disabled	No

Components
Manages the list of components that will be executed when the step is selected.

Name	Ext.	Size	Date
CC_HMI_Adv_7.6...	ucz	1 KB	6/3/2015 1:13 PM
CC_HMI_Adv_7.6...	bak	794 Byte	9/19/2012 10:56 AM
CC_HMI_Adv_7.6...	bak	840 Byte	6/3/2015 1:13 PM
CC_SO_2.6SP1	ucz	1 KB	6/3/2015 6:02 PM
CC_SO_2.6SP1.ucz	bak	1 KB	7/11/2012 2:56 PM
CC_SO_2.6SP1.u...	bak	1 KB	6/3/2015 1:13 PM

```
CHANDATA(1)
N20000 $MC_CHAN_NAME="TUTORIAL"
$MA_AX_VELO_LIMIT[0,AX1]?=2300
```

Identifier	Operator	Project/component	Step/task	Line
$MA_AX_VELO_LIMIT[0,AX5]	=	CP_MD.ucz	NC Channel 1	8
$MA_MAX_AX_VELO[AX5]	*	CP_MD.ucz	NC Channel 1	8
$MA_MAX_AX_VELO[AX$(UP.AXNr)]	=	CP_MD.ucz	NC Channel 1	18
$MC_CHAN_NAME	=	CP_MD.ucz	NC Channel 1	2

		No.	Message	Information
	⚠	200262	NCU area is not used	NCU data are to be processed, however the "Project/Package" register has been configured so that this data area will not be used.
C	✓	200001	Check component "CP_Alarm_SO_PCU", ok	Last changed: 6/3/2015 1:12:34 PM, z003b86x
C	✓	200001	Check component "CP_Alarm_HMI_Adv", ok	Last changed: 6/3/2015 1:12:33 PM, z003b86x
C	✓	200001	Check component "CP_MD", ok	Last changed: 6/3/2015 3:34:04 PM, z003b86x
C	✕	200001	Check component "CP_GUD", nok	Last changed: 6/3/2015 1:12:38 PM, z003b86x

✕ Finished Verification completed. 2 errors 3 warning messages occured

①	Project folder
②	Source explorer
③	Component editor
④	Cross references
⑤	Verification of package

4.2.2 Main menus and toolbars

Table 4-1

Project			
🗋	New...	<Ctrl+N>	Creates a new project under the specified name and storage path.
📂	Open...	<Ctrl+O>	Opens an existing project. Also permits an older project to be selected for conversion.

💾	Save	<Ctrl+S>	Saves the selected project and all internal or externally linked components belonging to this project, if there is a save request for these.		
💾	Save All	<Ctrl+Shift+S>	Saves all projects and components for which there is a save request.		
	Save As...		Saves **all** opened projects and components under a different name or location.		
🔒	Password protection		The project, the components, the package, and the logbook can be protected from unauthorized access with passwords.		
			Enter the password...		Log on to the open project using the password (extended rights).
			Edit password list...		Define the passwords with the corresponding rights on the package and project.
			Transfer password list		Transfer the created passwords to internal or external components.
✕	Display cross references...	<Ctrl+Alt+Shift+R>	Displays the cross references for the project and all of its components.		
	Verification run	<Ctrl+Shift+B>	Checks the project including all components. The verification run is started automatically before each deployment.		
	Deployment		Checks the project and deploys it as package.		
			Deploy Linux package (NCU)	USZ	Can be executed on the NCU
			Deploy Windows package (PC/PCU)	EXE	Can be executed on a PC or the PCU
			Deploy package without the executable part	UAZ	Cannot be executed
	Import Package...		Imports all data into a new project from a deployed package such as USZ, EXE, UAZ.		
✕	Delete Projects...		Allows the deletion of one or several projects of the current project archive that have not been opened. The project file and the project folder with all internal components are deleted.		
	Properties...		Shows important project properties, such as path of the project, date of change, changed by, version and comment.		
	CMC Report...		CMC report shows the properties of the project and all components in a dialog box.		
	Last Opened		Provides a fast entry to open the last projects used.		
	Close All		Closes the project and all of the components that have been opened.		
	Exit	<Alt+F4>	Exits the Expert program.		
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.		
🔄	Refresh	<F5>	Refreshes the step tree completely. This function is particularly for the refreshing of linked steps.		
View					

	Source explorer		Hides or displays the source explorer at the upper edge of the working area.	
	Cross references		Shows all identifiers, folders, and file objects used in the project on tabs "Identifiers" and "Folders and File objects".	
	Verification run		Hides or displays the verification of package at the lower edge of the application window.	
	Go to error/warning forward	<F8>	After a verification run, a jump can be made to the next/previous error or alarm entry.	
	Go to error/warning backward	<Shift+F8>		
	Toolbars	Standard	Allows toolbars to be hidden or displayed.	
		Quick search		
Component				
	New...	<Alt+Shift+N>	Creates a new component. Internal and external components can be created.	
	Open...	<Alt+Shift+O>	Opens an existing component.	
	Save	<Alt+Shift+S>	Saves the current component.	
	Save As...		Saves the current component under another name - either internally or externally.	
	Close	<Ctrl+F4>	Closes the current component.	
	Password protection		The current component is protected from unauthorized access by passwords.	
			Enter the password...	Log on in the current component (active tab) with another password (extended rights).
			Edit password list...	Define passwords with the corresponding rights in the current component.
			Transfer password list	Transfer the created passwords to internal or external components.
	Properties...		Shows important component properties, such as path of the component, date of change, changed by, version and comment.	
	CMC Report...			
	Manage (copy, delete, etc.)...	<Shift+F12>	Opens the "Manage" dialog box in which components can be copied, moved, structured and deleted.	
	Display non-linked components...		Determines all internal components of the currently opened project that are not linked – and allows these to be deleted.	
	Display cross references	<Ctrl+Alt+Shift+R>	Shows a list of all identifiers, folders, and file objects used in the current component.	
	Verification of package	<Ctrl+Shift+B>	Checks the current component. For an opened project, this is checked against its configuration.	
Component				
	Store in file...		Enables the storing of the current configuration for areas to be selected, in a file.	
	Read from file...		Imports configurations for areas to be selected, from a file. The current configuration is overwritten by the imported configuration data.	
Tools				
	Options...		Opens a dialog box in which the options can be set.	

	Adapt links...	in the project...	Links in the project and components can be adapted.
		in component...	
		in components of a storage folder...	
	Convert	from components of a storage folder...	Enables components from an older Expert version to be converted to the current Expert version in a folder.
		Delete backup files (*.bak)	Deletes the backup files created during the conversion.
Window			
⊟	New horizontal tab group		Displays the selected component in the new horizontal tab group.
◫	New vertical tab group		Displays the selected component in the new vertical tab group.
	Move to previous tab group		Moves the selected component to the previous tab group.
	Move to next tab group		Moves the selected component to the next tab group.
	Note Window Layout		Note the layout of the user interface areas for the next start of the program.
Help			
?	Contents	<F1>	Displays the online help of the **Expert** program which is identical to this manual.
	About...		Contains information on the version and copyright.

NOTICE

Data can no longer be found

If projects, components, and associated data are moved more than once without storage in a buffer, automatic path adaptation may no longer be possible. Components or linked data can then no longer be found.

You should therefore run a verification of projects every time after packages or components are moved.

When projects and components are copied with unfavorable conditions for automation path adaptation, the project may access the original components or linked data.

4.2.3 Source Explorer

A source explorer is integrated in Expert which is used similarly to the Windows Explorer.
Only read operations are offered for the source explorer in order to avoid unintentional deletion, moving or changing of data.
An external file manager can also be used to copy files and folders to the components, or to link these.

Copy

Folders and files can be dragged from the source explorer and dropped in a component. Copying is performed when the <Ctrl> key is pressed. Linking is performed when the <Ctrl +Shift> keys are pressed. Otherwise you are prompted as to what should be done.
Copy & paste is also possible.

Refresh

The <F5> key refreshes the display of the source explorer.

Double-click

A double-click on the title bar minimizes the source explorer or restores it.
When a linked file is double-clicked, the source explorer changes to its path.

4.2.4 Converting projects from older CMC versions

When opening a project or a component, Expert checks the existing CMC version. If the CMC version of the project to be opened is earlier than the installed CMC version, Expert converts the project and the internally and externally linked components.

The conversion only takes place after a prompt, which must be confirmed by the user.

The prompt also specifies the CMC version that was used to create the project to be opened and allows the creation of *.bak files before the conversion.
The *.bak files can be subsequently deleted via the main menu "Tools > Convert > Delete backup files (*.bak)".

It is recommended that each project be manually backed up or archived with all of the associated internal and external components before the conversion. It is also recommended that a conversion only be carried out from the previous to the current CMC version, i.e. no CMC versions should be skipped.

4.3 Project folder

4.3.1 Package

The basic settings for the package are made on the Package tab.

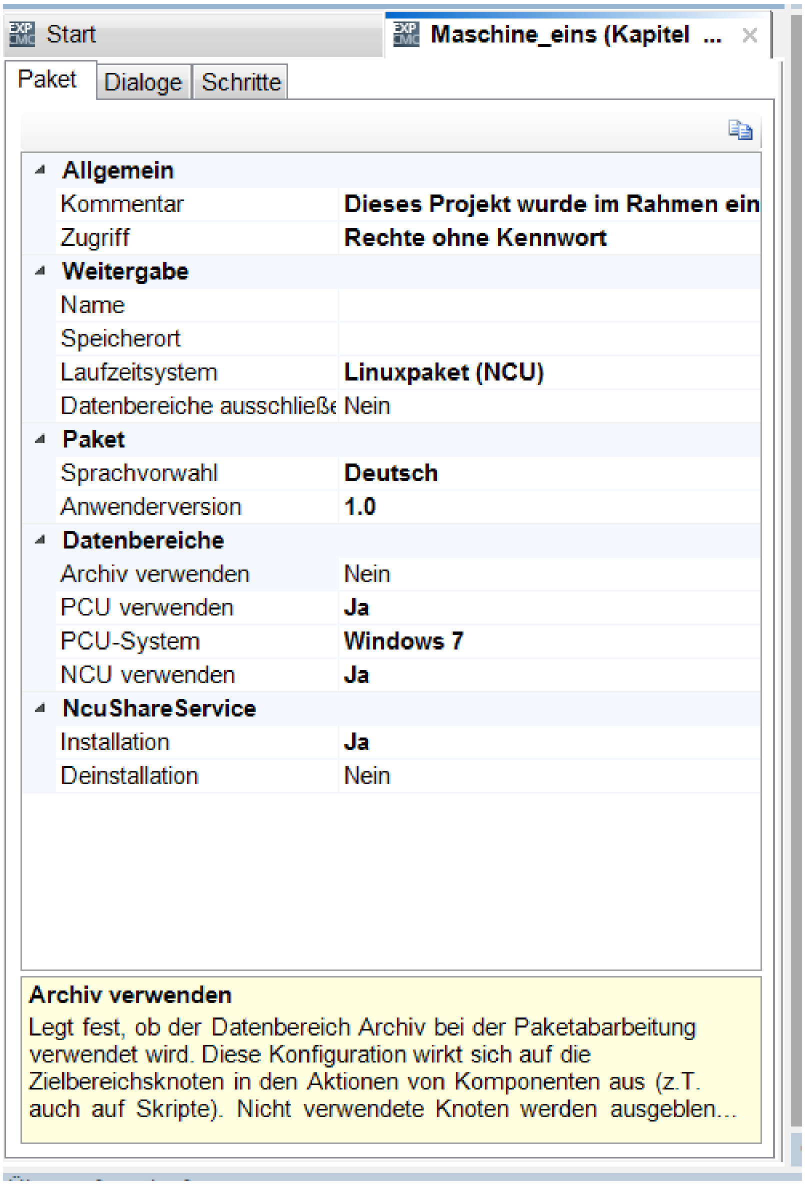
After specifying the project properties, all the relevant data for the deployment of the project is configured.

Then the settings for the package and NcuShareService are defined and an executable package is created.

Specifying the configuration for Package

You can configure the following property groups on the "Package" tab:

- General
- Deployment
- Package
- Data areas
- NcuShareService



Allgemein	
Kommentar	**Dieses Projekt wurde im Rahmen ein**
Zugriff	**Rechte ohne Kennwort**
Weitergabe	
Name	
Speicherort	
Laufzeitsystem	**Linuxpaket (NCU)**
Datenbereiche ausschließe	Nein
Paket	
Sprachvorwahl	**Deutsch**
Anwenderversion	**1.0**
Datenbereiche	
Archiv verwenden	Nein
PCU verwenden	**Ja**
PCU-System	**Windows 7**
NCU verwenden	**Ja**
NcuShareService	
Installation	**Ja**
Deinstallation	Nein

Archiv verwenden

Legt fest, ob der Datenbereich Archiv bei der Paketabarbeitung verwendet wird. Diese Konfiguration wirkt sich auf die Zielbereichsknoten in den Aktionen von Komponenten aus (z.T. auch auf Skripte). Nicht verwendete Knoten werden ausgeblen...

Figure 4-1 Project / Package tab - project engineer view

Note

Help areas for Project / Package tab

Note the color-coded help area under Project / Package.

Click on a property to display the relevant description.

Display all system variables via a dialog button

Via the button (Display all system variables via a dialog), you can call the "Package system variables" dialog. All system variables that can be used in Package are listed here.

Information on the Package system variables can be found in the System variables for package properties Up.$Pack (Page 272) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

General

The following settings are available:

Table 4-2 General

Comment	Project-internal information can be entered here. This comment is not displayed during package processing or in the logbook.
Access	Access rights are displayed and defined here. Click the button to open the "Edit password list [Rights without a password]" dialog. More detailed information is provided at "Editing password protection (Page 240)". General information on the password protection (Page 240).

Deployment

The following settings are available:

Table 4-3 Deployment

Name	Here, you define the name of the package for this deployment.
Storage location	Here, you define the storage location, i.e. the absolute path under which the package is saved.

Runtime system	LINUX package (NCU) Windows package (PC/PCU) Package without the executable part	Here, you select the runtime system on which the package is executed.
Exclude data areas	Yes No	Here, you define whether data from unused data areas Archive, NCU, and PCU is excluded from deployment. The data of the non-selected PCU system is also excluded. The size of the package can be reduced by excluding the data areas.

Package

The following settings are available:

Table 4-4 Package

Language preselection	German English	Here, you select the user interface language for processing the package. The language can be switched for each dialog separately by the user during the package execution. The user fields (yellow fields) remain unchanged.
User version	1.0	The package can be versioned by the user via the user version. An environment variable of the type Up. $Pack.ProdVersion = "1.0" can also be entered here.

Data areas

The following settings are available:

Table 4-5 Data areas

Use archive	Yes No	Here, you specify whether the Archive data area is used for package execution. This configuration affects the target area nodes in the actions of components (partially also scripts). Unused nodes are hidden or highlighted in color if they contain data.
Use PCU	Yes No	Here, you specify whether the PCU data area is used for package execution. This configuration affects the target area nodes in the actions of components (partially also scripts). Unused nodes are hidden or highlighted in color if they contain data.

PCU system	Windows XP Windows 7	Here, you select the operating system of the PCU.
Use NCU	Yes No	Here, you specify whether the NCU data area is to be used for the package execution. This configuration affects the target area nodes in the actions of components (partially also scripts). Unused nodes are hidden or highlighted in color if they contain data.

NcuShareService

The following settings are available:

Table 4-6 NcuShareService

Installation	Yes No	The NcuShareService is installed on the "NCU access" dialog page if it has not been installed yet or an older versions exists on the PC/PCU. The service is only required if a Windows package is to be executed with NCU data area.
Uninstallation	Yes No	The NcuShareService is uninstalled after executing the package. The uninstall can only be performed if the installed service has the same service version in the current package.

Further information on the NcuShareService can be found in the Create MyConfig NcuShare (Page 347) section or in the online help of the same name.

See also

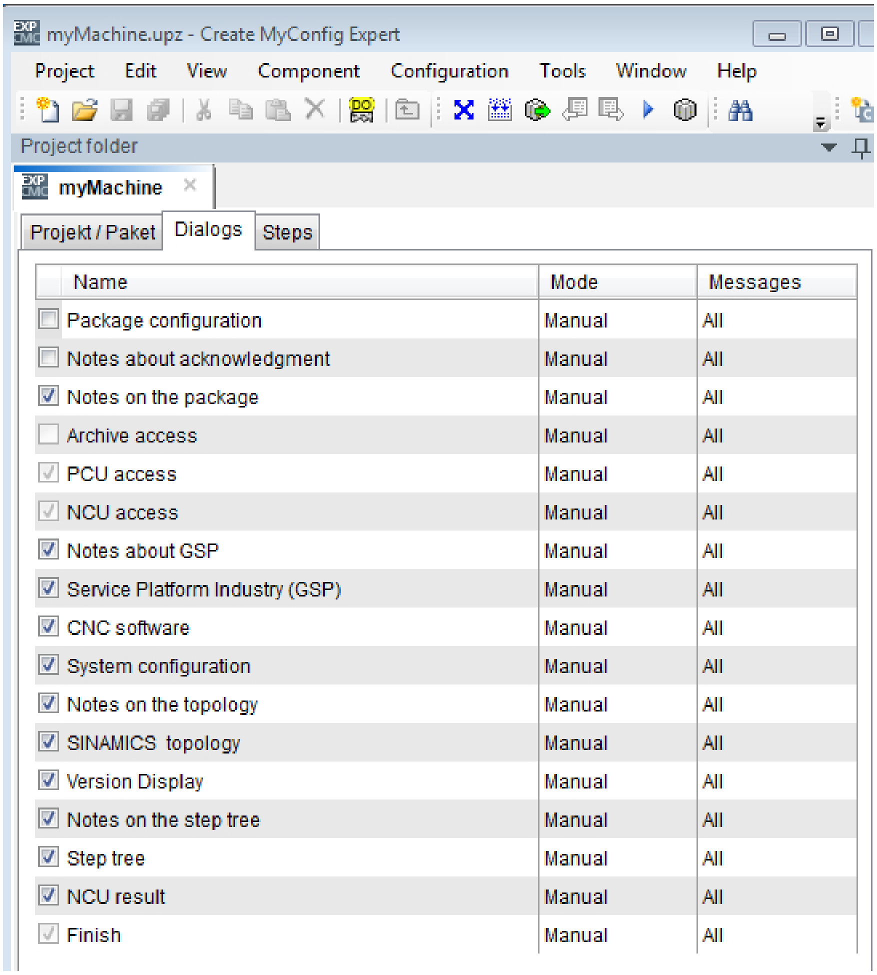
Editing a password list (Page 242)

4.3.2 Dialogs

Overview of dialog pages

Below the project map in the "Dialogs" tab, you can activate the dialogs to be executed and configure their contents and behavior.

You can see the dialog list with the dialog pages that are possible to configure in the following:



	Name	Mode	Messages
□	Package configuration	Manual	All
□	Notes about acknowledgment	Manual	All
☑	Notes on the package	Manual	All
□	Archive access	Manual	All
✓	PCU access	Manual	All
✓	NCU access	Manual	All
☑	Notes about GSP	Manual	All
☑	Service Platform Industry (GSP)	Manual	All
☑	CNC software	Manual	All
☑	System configuration	Manual	All
☑	Notes on the topology	Manual	All
☑	SINAMICS topology	Manual	All
☑	Version Display	Manual	All
☑	Notes on the step tree	Manual	All
☑	Step tree	Manual	All
☑	NCU result	Manual	All
✓	Finish	Manual	All

Figure 4-2 Dialogs tab - Overview

Defining general properties for all dialogs

Generally applicable properties can be defined for all dialogs.

Name

In the table column under "Name", dialog pages with a black checkbox can be activated or deactivated for "Editing" directly in the dialog list or via the property "Edit" in the property field.

Dialog pages with a gray checkbox are **always** required to execute the package, in accordance with the configuration of the areas in the Project/Package tab, and **cannot** be deselected.

Editing mode for all dialogs

By right-clicking on one of the table columns (Name, Mode or Messages) or on a dialog page, you can uniformly define the Mode property for all dialogs: "Editing mode for all dialogs".

The following options are offered and are activated when the package is executed:

Manual	The dialog page is completely displayed, the operator can make changes and must trigger the dialog page advance or execution.
Automatic	The dialog page is completely displayed, however, the operator **cannot** make any changes and it is executed automatically.
Progress	Processing is analogous to the Automatic mode without the configuration information being displayed.

Display messages for all dialogs

Also by right-clicking on one of the table columns (Name, Mode or Messages) or on a dialog page, you can uniformly define the Messages property for all dialogs: "Display messages for all dialogs".

The following options are available:

Errors	The "Display messages for all dialogs" property allows messages to be suppressed during execution. Messages to be suppressed are acknowledged via automatic actuation of the standard button.
Errors and warnings	
All	

The following table shows an overview of when which message is displayed or not displayed.

Message	All	Errors and warnings	Errors
System error	On	On	On
Run error	On	On	On
Run warning	On	On	Off
Message when the step script is fulfilled	On	Off	Off
Warning for activated/ not activated step	On	On	Off
Error function	On	On	On
Warning function	On	On	Off
Msg function	On	Off	Off
Input functions	On	On	On

Using the dialog list, you can centrally view the editing mode and message display for all dialogs, and select individually or for all dialogs (context menu).

Dialog pages for the Dialogs tab

The following dialog pages are run through chronologically during package execution.

In the following sections, the property groups and the setting options for the respective dialog pages are described.

Finally, the configured dialog page is shown how it is displayed for the operator or commissioning engineer during package execution.

All configurable dialog pages have multiple property groups. All dialog pages have the following two property groups:

- Dialog
- Operating instructions

These property groups are described only once in the "Notes about acknowledgement (Page 104)" dialog, but they apply for all dialog pages.

To the dialog pages

Here you can find detailed information about the individual dialog pages:

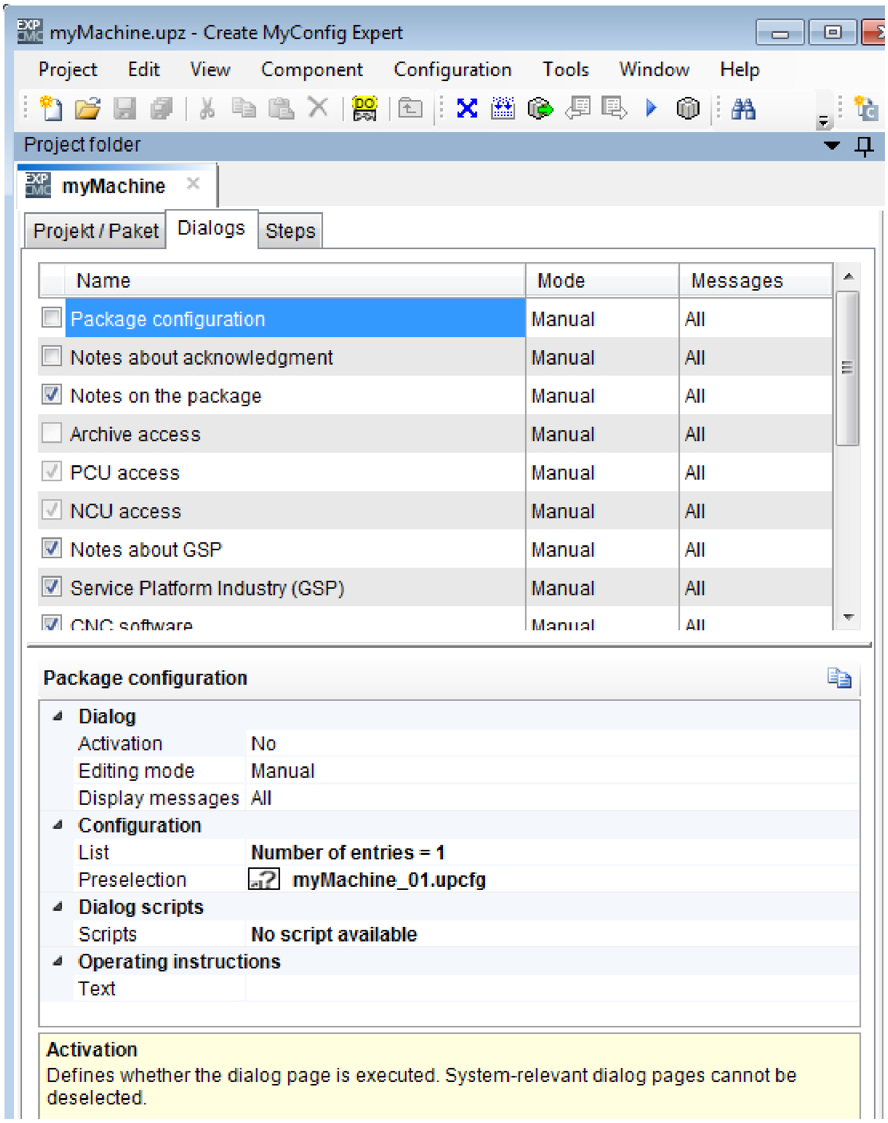
- Package configuration (Page 99)
- Notes about acknowledgement (Page 104)
- Notes on the package (Page 107)
- Archive access (Page 109)
- PCU access (Page 112)
- NCU access (Page 115)
- Notes on GSP (Service Platform Industry) (Page 119)
- Service Platform Industry (GSP) (Page 121)
- CNC software (Page 126)
- System configuration (Page 131)
- Notes on the topology (Page 141)
- SINAMICS topology (Page 143)
- Version display (Page 150)
- Notes on the step tree (Page 153)
- Step tree (Page 155)
- NCU result (Page 159)
- End (Page 163)

4.3.2.1 Dialog page: Package configuration

Configuration

You can configure the following property groups for the "Package configuration" dialog page:

- Dialog
- Configuration
- Dialog scripts



Name	Mode	Messages
□ Package configuration	Manual	All
□ Notes about acknowledgment	Manual	All
☑ Notes on the package	Manual	All
□ Archive access	Manual	All
☑ PCU access	Manual	All
☑ NCU access	Manual	All
☑ Notes about GSP	Manual	All
☑ Service Platform Industry (GSP)	Manual	All
☑ CNC software	Manual	All

Package configuration

Dialog	
Activation	No
Editing mode	Manual
Display messages	All
Configuration	
List	Number of entries = 1
Preselection	myMachine_01.upcfg
Dialog scripts	
Scripts	No script available
Operating instructions	
Text	

Activation
Defines whether the dialog page is executed. System-relevant dialog pages cannot be deselected.

Figure 4-3 Package configuration - project engineer view

Display all system variables via a dialog button

You open the "Package configuration - System variables" dialog via the button 📋 (display all system variables via a dialog).
All system variables to be used under "Package configuration" are listed here.

Information on the system variables·can be found in the System variables for the Package configuration dialog (Page 274) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog

More detailed information about the property group "Dialog" can be found under the dialog page: Notes about acknowledgement (Page 104).

Configuration

The following settings are available:

List	The list of all existing package configurations in the package is displayed in the "List of package configurations".
Preselection	Here, you make the preselection for the selected package configuration for package execution.

Operating instructions

The following settings are available:

Text	

Dialog scripts

Under "Dialog scripts", entry "Scripts", via the button … you open the dialog page "ScriptEditor dialog - <Name of the opened dialog page> (<Time of execution>)". You can edit scripts here that are to be executed at selectable times during the package execution.

In this dialog, the specified dialog scripts of selected dialogs can be assigned to the desired events (=time) without having to call each individual dialog.

The dialogs are listed in the following table:

Table 4-7 Assignment dialog - Event

Dialog	Event
Package configuration	OnInit OnNext OnEnd
Archive access	
PCU access	
NCU access	
Service Platform Industry (GSP)	
CNC software	
System configuration	
SINAMICS topology	
Step tree	
NCU result	

Meaning of the events:

Table 4-8 Event - Meaning

Event	Meaning / Time of script execution
OnInit	The script is executed before the dialog page is displayed. OnInit is used: ● To skip this dialog page ● To change settings of this dialog page because of the current sequence (e.g. settings for write access to the system variables can still be changed effectively)
OnNext	The script is executed in the dialog immediately after clicking the "Next" button. OnNext is used: ● To check operator inputs after clicking the "Next" key ● To return to the input At this point in time, set values of the dialog page can be checked by exporting system variables. The dialog execution can be interrupted in the script with the Redo() function. In this case, settings of the dialog page can be corrected. The execution of the dialog page can be restarted. The script "OnNext" is also executed again.
OnEnd	After successful processing, the script is executed at the end of a dialog page. Here, via write access to system variables for example, you can set which of the following dialogs is to be executed or not.

Inserting a script

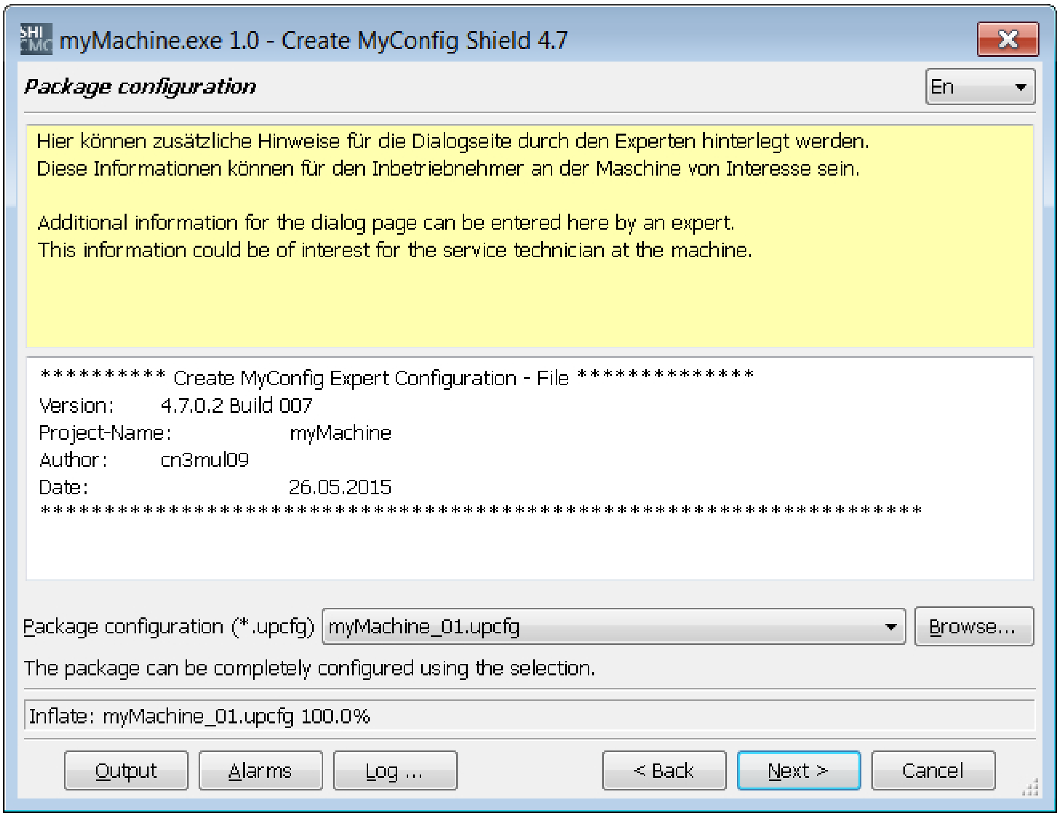
Information on inserting a script and on the following options can be found in the Scripting (Page 197) section at:

- Sections (Page 204)
- Area data (Page 206)
- Control structures (Page 206)
- Operators (Page 209)
- Functions (Page 212)
- Procedures (Page 231)

Dialog view during the package execution (for commissioning engineers / operators)

Description

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



Hier können zusätzliche Hinweise für die Dialogseite durch den Experten hinterlegt werden.
Diese Informationen können für den Inbetriebnehmer an der Maschine von Interesse sein.

Additional information for the dialog page can be entered here by an expert.
This information could be of interest for the service technician at the machine.

```
********** Create MyConfig Expert Configuration - File **************
Version:        4.7.0.2 Build 007
Project-Name:                myMachine
Author:         cn3mul09
Date:                        26.05.2015
*******************************************************************
```

Package configuration (*.upcfg) myMachine_01.upcfg

The package can be completely configured using the selection.

Inflate: myMachine_01.upcfg 100.0%

Figure 4-4 Package configuration dialog - commissioning engineer / operator view

Buttons for the dialogs

Table 4-9 Meaning of 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	The alarm list is shown or hidden. It displays errors and messages from NC, PLC, drive, and HMI. It approximately corresponds to the "Diagnostics" used in SINUMERIK Operate.
Log…	This button opens the CMC logbook for package execution.
Back / Next	These buttons lead through the dialog pages accordingly.
Cancel	The execution of the package is canceled. A prompt generally appears. This prompt must be acknowledged. After a dialog is canceled, the "Finish" dialog is always displayed, thus allowing the logbook to be backed up. The cancellation process can take place immediately or after a time delay, depending on the dialog page and the execution status.

Note

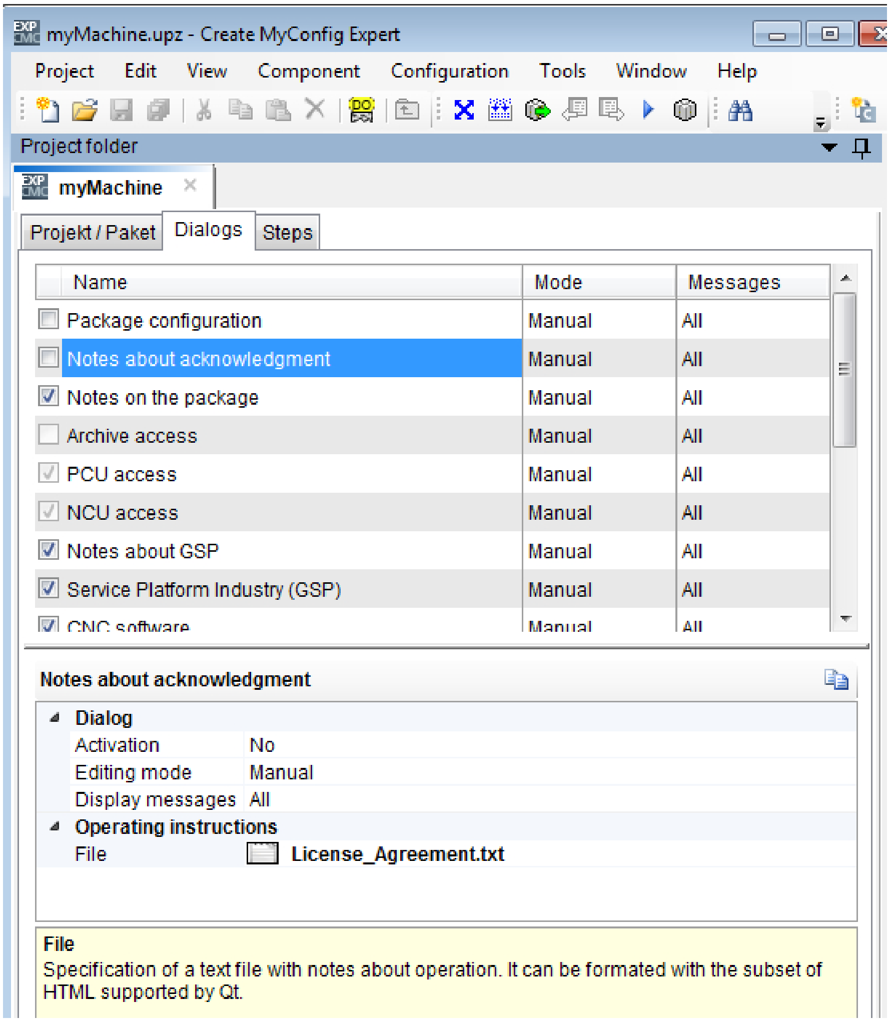
The buttons are identical for all dialog pages and thus have the same meaning.

4.3.2.2 Dialog page: Notes about acknowledgement

Configuration

You can configure the following property groups for the "Notes about acknowledgement" dialog page:

- Dialog
- Operating instructions



	Name	Mode	Messages
☐	Package configuration	Manual	All
☐	Notes about acknowledgment	Manual	All
☑	Notes on the package	Manual	All
☐	Archive access	Manual	All
☑	PCU access	Manual	All
☑	NCU access	Manual	All
☑	Notes about GSP	Manual	All
☑	Service Platform Industry (GSP)	Manual	All
☑	CNC software	Manual	All

Notes about acknowledgment

Dialog	
Activation	No
Editing mode	Manual
Display messages	All
Operating instructions	
File	**License_Agreement.txt**

File
Specification of a text file with notes about operation. It can be formated with the subset of HTML supported by Qt.

Figure 4-5 Notes about acknowledgement - project engineer view

Note

Help area below the Dialogs tab

Note the yellow help area under the Dialogs tab. Click on a property to display the relevant description.

Display all system variables via a dialog button

You open the "Notes about acknowledgement - System variables" dialog via the button 📋 (display all system variables via a dialog). All system variables to be used under "Notes about acknowledgement" are listed here.

Information on the system variables can be found in the System variables for the Notes on confirmation dialog (Page 274) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog

You have the capability of centrally defining the "Dialog" properties for all of the dialogs together, see the Summary "Defining general properties for all dialogs (Page 96)" section.

However, you can also individually adapt the "Dialog" properties for each dialog page, which should then take effect when the package is executed:

Activation	Yes	Here, you specify whether the dialog page is to be used for the package execution. If this is not activated for the package execution, the following properties have no effect.
	No	
Editing mode	Manual	The dialog page is completely displayed, the operator can make changes and must trigger the dialog page advance or execution.
	Automatic	The dialog page is completely displayed and executed automatically. The operator **cannot** make any changes.
	Progress	Processing is analogous to the "Automatic" mode without the configuration information being displayed. A progress indicator is displayed in a small display window.
Display messages	Errors	The "Display messages" property allows messages to be suppressed during the package execution. Messages to be suppressed are acknowledged via automatic actuation of the standard button.
	Errors and warnings	
	All	

Operating instructions

User-specific "Operating instructions" can be configured for most dialog pages. The text can be formatted using HTML tags.

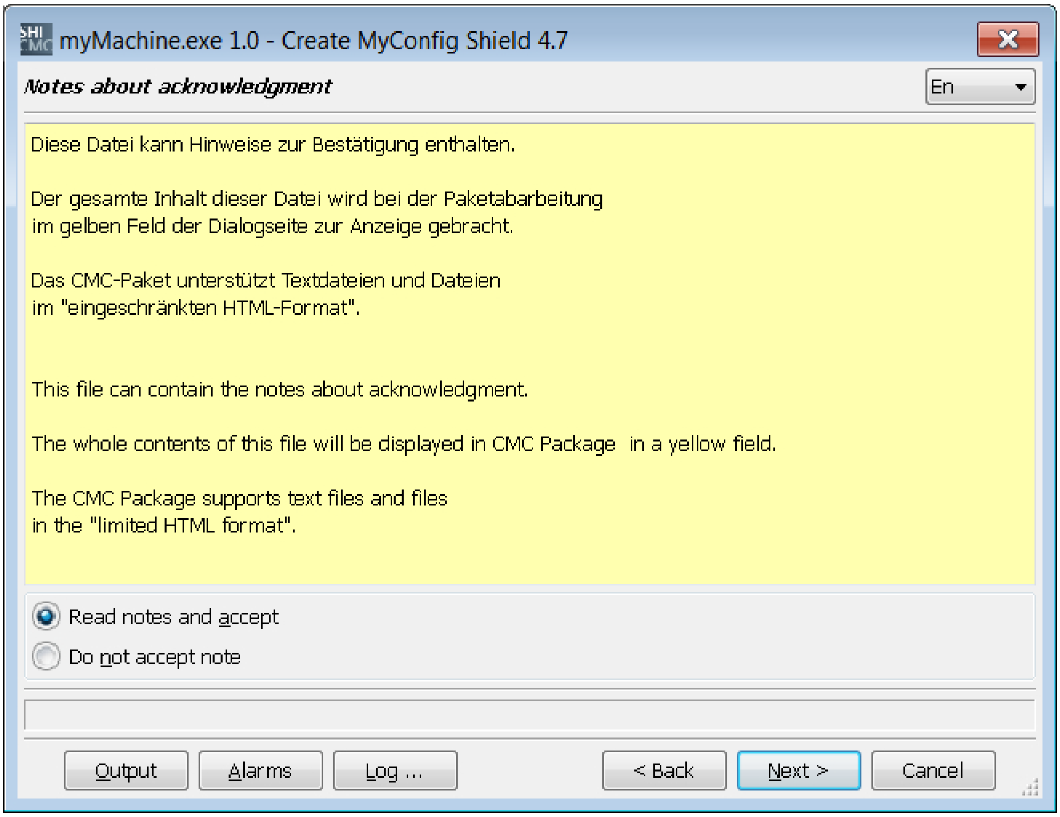
When executing, the operating instructions are displayed with a yellow background so that it is easier to distinguish them from fixed user interface texts.

Dialog view during the package execution (for commissioning engineers / operators)

On the "Notes about acknowledgement" dialog page, you can display your license terms from a file in text format or restricted HTML format.
You can also use this page to define that the user must confirm having read the displayed information before the package is executed.

Package execution can only be continued when the user agreement has been accepted.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).


myMachine.exe 1.0 - Create MyConfig Shield 4.7
Notes about acknowledgment
En

Diese Datei kann Hinweise zur Bestätigung enthalten.

Der gesamte Inhalt dieser Datei wird bei der Paketabarbeitung
im gelben Feld der Dialogseite zur Anzeige gebracht.

Das CMC-Paket unterstützt Textdateien und Dateien
im "eingeschränkten HTML-Format".

This file can contain the notes about acknowledgment.

The whole contents of this file will be displayed in CMC Package in a yellow field.

The CMC Package supports text files and files
in the "limited HTML format".

● Read notes and accept
○ Do not accept note

Figure 4-6 Notes about acknowledgement dialog - commissioning engineer / operator view

Buttons in dialogs

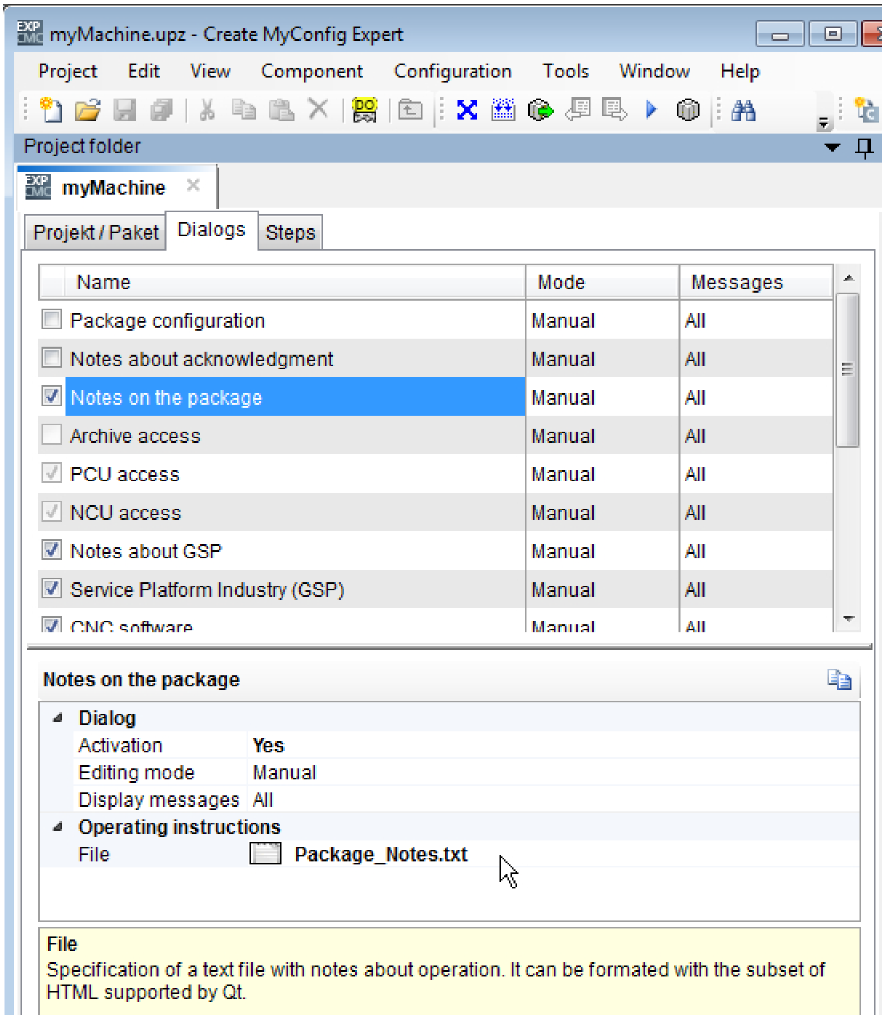
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.3 Dialog: Notes on the package

Configuration

You can configure the following property groups for the "Notes on the package" dialog page:

- Dialog
- Operating instructions



	Name	Mode	Messages
☐	Package configuration	Manual	All
☐	Notes about acknowledgment	Manual	All
☑	Notes on the package	Manual	All
☐	Archive access	Manual	All
☑	PCU access	Manual	All
☑	NCU access	Manual	All
☑	Notes about GSP	Manual	All
☑	Service Platform Industry (GSP)	Manual	All
☑	CNC software	Manual	All

Notes on the package

Dialog	
Activation	**Yes**
Editing mode	Manual
Display messages	All
Operating instructions	
File	**Package_Notes.txt**

File
Specification of a text file with notes about operation. It can be formated with the subset of HTML supported by Qt.

Figure 4-7 Notes on the package dialog - project engineer view

Display all system variables via a dialog button

You open the "Notes on the package - system variables" dialog via the button (display all system variables via a dialog). All system variables to be used under "Notes on the package" are listed here.

Information on the system variables·can be found in the System variables for the Notes on the package dialog (Page 274) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

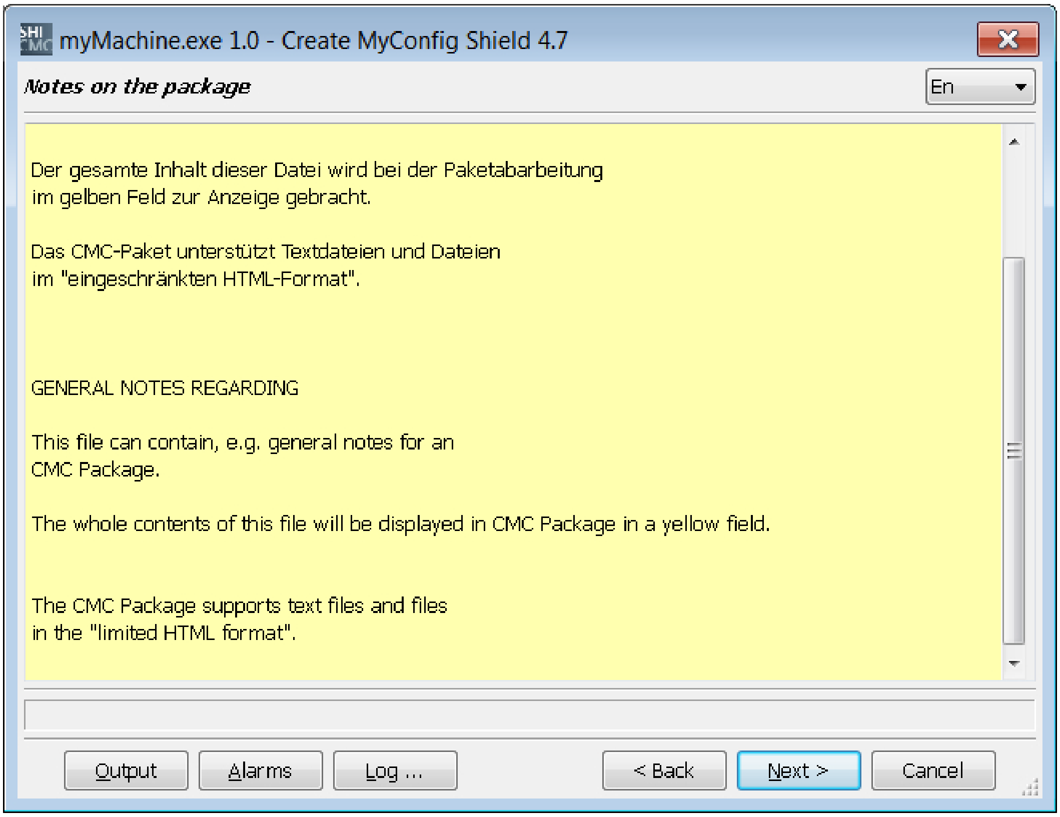
Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

Dialog view during the package execution (for commissioning engineers / operators)

A file in text format or restricted HTML format can be displayed with notes about executing the package on the "Notes on the package" dialog.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



myMachine.exe 1.0 - Create MyConfig Shield 4.7

Notes on the package

Der gesamte Inhalt dieser Datei wird bei der Paketabarbeitung
im gelben Feld zur Anzeige gebracht.

Das CMC-Paket unterstützt Textdateien und Dateien
im "eingeschränkten HTML-Format".

GENERAL NOTES REGARDING

This file can contain, e.g. general notes for an
CMC Package.

The whole contents of this file will be displayed in CMC Package in a yellow field.

The CMC Package supports text files and files
in the "limited HTML format".

Figure 4-8 Notes on the package dialog - commissioning engineer / operator view

Buttons in dialogs

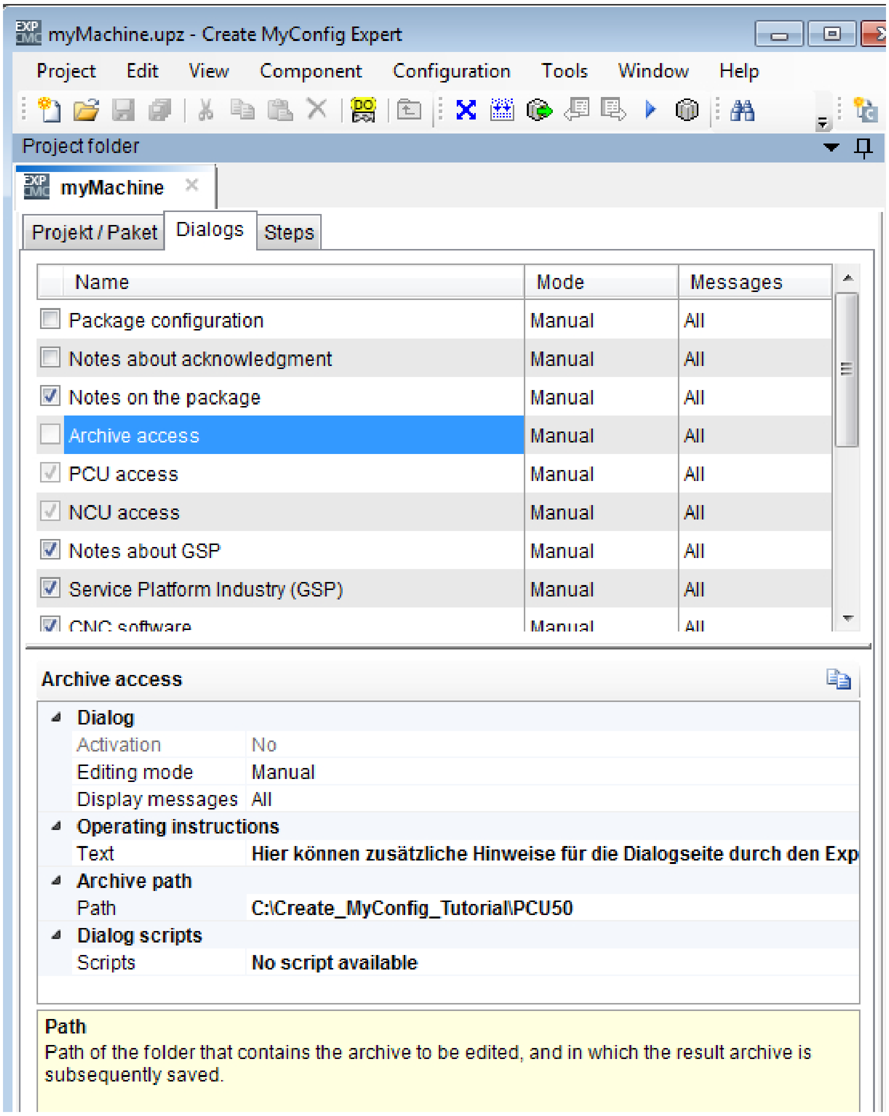
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.4 Dialog page: Archive access

Configuration

You can configure the following property groups for the "Archive access" dialog page:

- Dialog
- Operating instructions
- Archive path
- Dialog scripts



Name	Mode	Messages
☐ Package configuration	Manual	All
☐ Notes about acknowledgment	Manual	All
☑ Notes on the package	Manual	All
☐ Archive access	Manual	All
☑ PCU access	Manual	All
☑ NCU access	Manual	All
☑ Notes about GSP	Manual	All
☑ Service Platform Industry (GSP)	Manual	All
☑ CNC software	Manual	All

Archive access

Dialog	
Activation	No
Editing mode	Manual
Display messages	All
Operating instructions	
Text	**Hier können zusätzliche Hinweise für die Dialogseite durch den Exp**
Archive path	
Path	**C:\Create_MyConfig_Tutorial\PCU50**
Dialog scripts	
Scripts	**No script available**

Path
Path of the folder that contains the archive to be edited, and in which the result archive is subsequently saved.

Figure 4-9 Archive access - project engineer view

The "Archive access" dialog page is activated if "Use archive" was selected in the "Project / Package" tab under Packages > Data areas.

The package configuration is only possible without the areas NCU and PCU.

Display all system variables via a dialog button

You open the "Archive access - System variables" dialog via the button 📋 (display all system variables via a dialog). All system variables to be used under "Archive access" are listed here.

Information on the system variables·can be found in the System variables for the Archive access dialog (Page 275) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

Archive path

Under the specified "Path" there is the folder which contains the archives to be executed and into which the result archive is re-saved.

Dialog scripts

Under "Dialog scripts", entry "Scripts", via the button ... you open the "ScriptEditor dialog - Archive access (<Time of execution>)" dialog page.
You can edit scripts here that are to be executed at selectable times during the package execution.

In this dialog, the specified dialog scripts of selected dialogs can be assigned to the desired events (=time) without having to call each individual dialog.

Assignment dialog - Event

The dialogs in which dialog scripts can be assigned to selected times (events), are listed in the Assignment dialog - Event (Page 101) table.

Meaning of the events:

Possible events and their meaning can be found in the Event - Meaning (Page 102) table.

Inserting a script

Information on inserting a script and on the following options can be found in the Scripting (Page 197) section at:

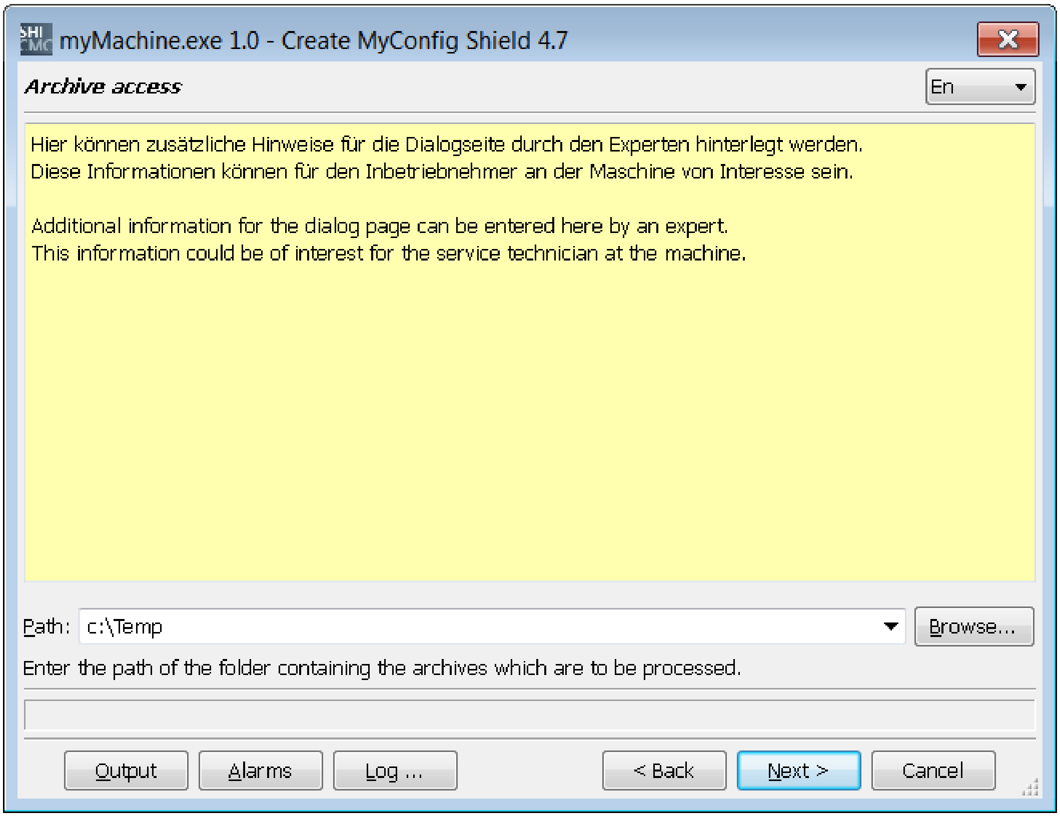
- Sections (Page 204)
- Area data (Page 206)
- Control structures (Page 206)
- Operators (Page 209)

- Functions (Page 212)
- Procedures (Page 231)

Dialog view during the package execution (for commissioning engineers / operators)

Execution of the "Archive access" dialog page cannot be deactivated. It is always automatically executed when executing a package under Windows that addresses either the "Archive area" or "PCU area".

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



Hier können zusätzliche Hinweise für die Dialogseite durch den Experten hinterlegt werden.
Diese Informationen können für den Inbetriebnehmer an der Maschine von Interesse sein.

Additional information for the dialog page can be entered here by an expert.
This information could be of interest for the service technician at the machine.

Path: c:\Temp

Enter the path of the folder containing the archives which are to be processed.

Figure 4-10 Archive access dialog - commissioning engineer / operator view

Buttons in dialogs

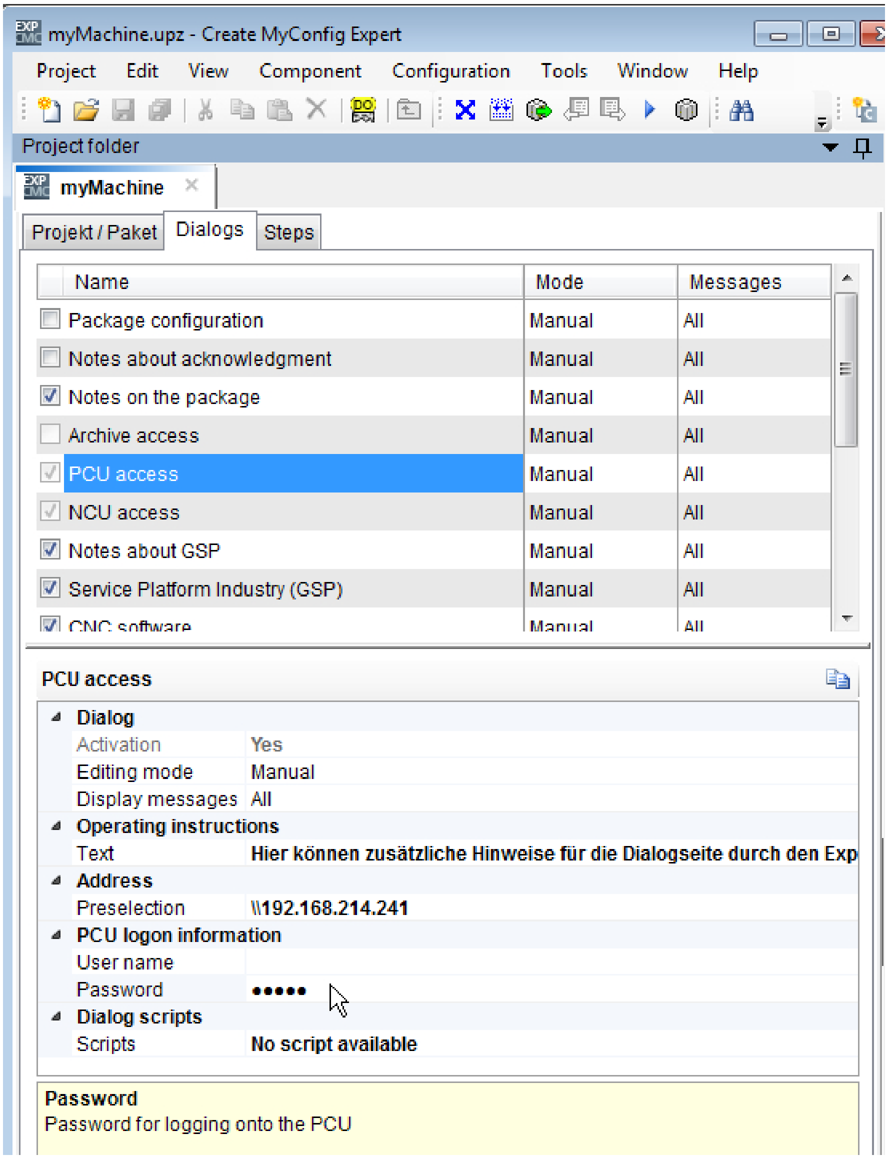
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.5 Dialog page: PCU access

Configuration

You can configure the following property groups for the "PCU access" dialog page:

- Dialog
- Operating instructions
- Address
- PCU login information
- Dialog scripts



	Name	Mode	Messages
□	Package configuration	Manual	All
□	Notes about acknowledgment	Manual	All
☑	Notes on the package	Manual	All
□	Archive access	Manual	All
☑	PCU access	Manual	All
☑	NCU access	Manual	All
☑	Notes about GSP	Manual	All
☑	Service Platform Industry (GSP)	Manual	All
☑	CNC software	Manual	All

PCU access

Dialog	
Activation	Yes
Editing mode	Manual
Display messages	All
Operating instructions	
Text	Hier können zusätzliche Hinweise für die Dialogseite durch den Exp
Address	
Preselection	\\192.168.214.241
PCU logon information	
User name	
Password	•••••
Dialog scripts	
Scripts	No script available

Password
Password for logging onto the PCU

Figure 4-11 PCU access - project engineer view

Display all system variables via a dialog button

You open the "PCU access - System variables" dialog via the button (display all system variables via a dialog). All system variables to be used under "PCU access" are listed here.

Information on the system variables can be found in the System variables for the PCU access dialog (Page 276) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

Address

The "Preselection" is already set to the default address of the "PCU50" 192.168.214.241 during configuration.

Note

If the package is to be executed locally on a PCU, the specified address has no significance.

PCU login information

The following settings are available for the "PCU login information" property:

User name	Neither a user name nor a password is pre-assigned for a new project.
Password	If the user name and password are not specified, a logon is started with the standard user of the PCU50 XP "auduser" and the appropriate password for the package execution.

Dialog scripts

Under "Dialog scripts", entry"Scripts", the [...] button calls the "ScriptEditor dialog - PCU access (<Time of the execution>)" dialog page.
You can edit scripts here that are to be executed at selectable times during the package execution.

In this dialog, the specified dialog scripts of selected dialogs can be assigned to the desired events (=time) without having to call each individual dialog.

Assignment dialog - Event

The dialogs in which dialog scripts can be assigned to selected times (events), are listed in the Assignment dialog - Event (Page 101) table.

Meaning of the events:

Possible events and their meaning can be found in the Event - Meaning (Page 102) table.

Inserting a script

Information on inserting a script and on the following options can be found in the Scripting (Page 197) section at:

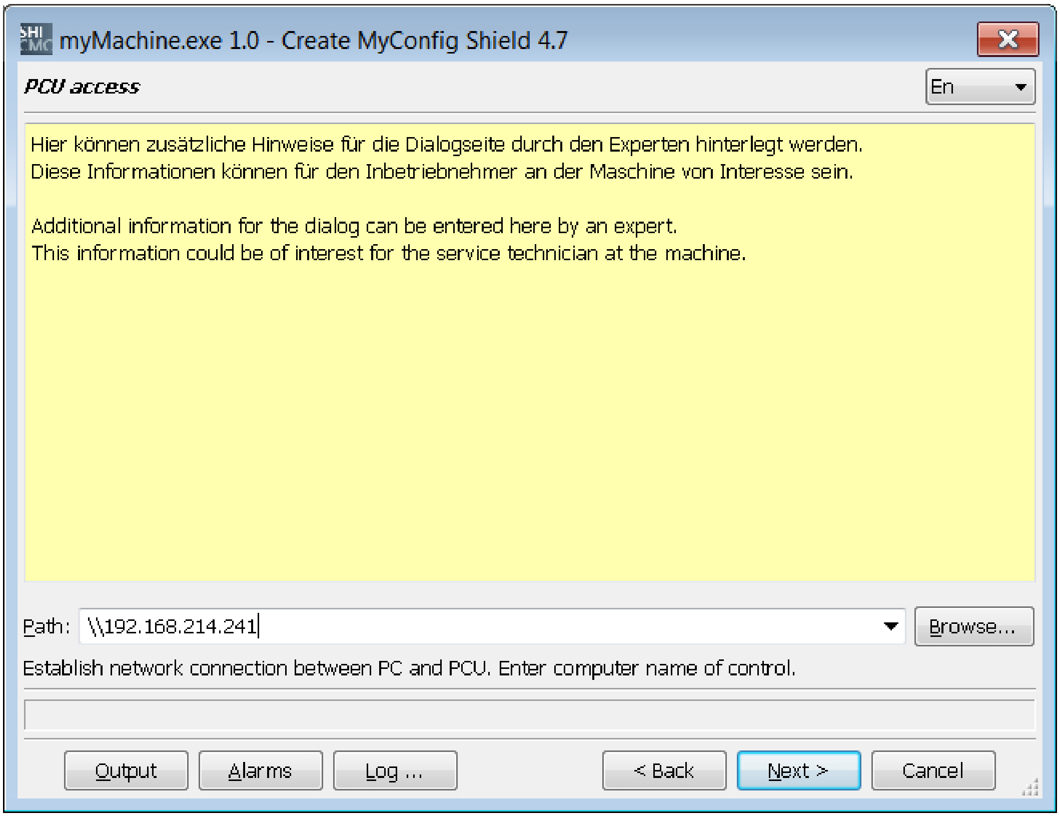
- Sections (Page 204)
- Area data (Page 206)
- Control structures (Page 206)
- Operators (Page 209)
- Functions (Page 212)
- Procedures (Page 231)

Dialog view during the package execution (for commissioning engineers / operators)

Execution of the "PCU access" dialog page cannot be deactivated.

If at "Package > Data areas > Use PCU" has been selected in the Project / Package tab, this dialog page is executed automatically when executing a package.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



myMachine.exe 1.0 - Create MyConfig Shield 4.7

PCU access

Hier können zusätzliche Hinweise für die Dialogseite durch den Experten hinterlegt werden.
Diese Informationen können für den Inbetriebnehmer an der Maschine von Interesse sein.

Additional information for the dialog can be entered here by an expert.
This information could be of interest for the service technician at the machine.

Path: \\192.168.214.241

Establish network connection between PC and PCU. Enter computer name of control.

Figure 4-12 PCU access dialog - commissioning engineer / operator view

Buttons in dialogs

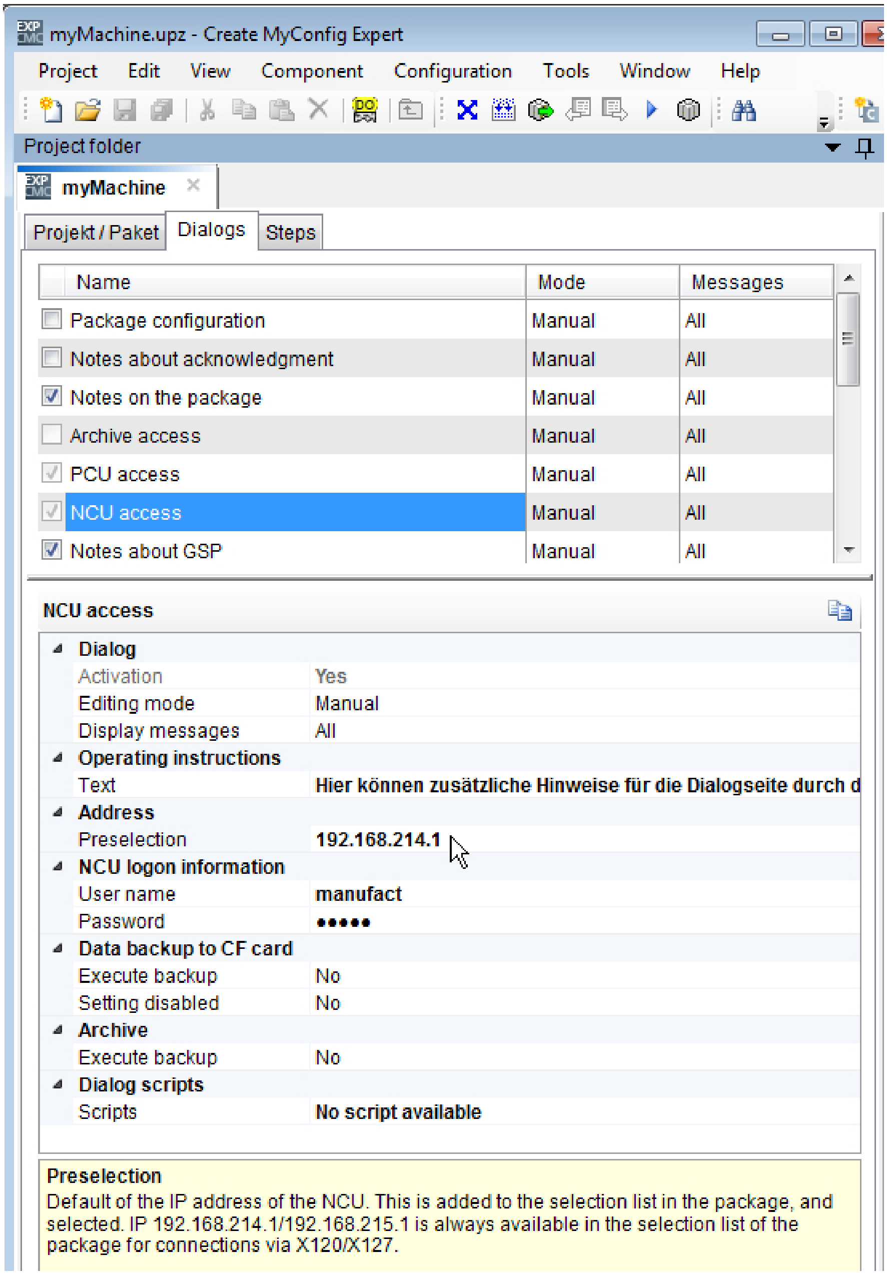
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.6 Dialog page: NCU access

Configuration

You can configure the following property groups for the "NCU access" dialog page:

- Dialog
- Operating instructions
- Address
- NCU login information
- Data backup CF card
- Archive
- Dialog scripts



	Name	Mode	Messages
□	Package configuration	Manual	All
□	Notes about acknowledgment	Manual	All
☑	Notes on the package	Manual	All
□	Archive access	Manual	All
☑	PCU access	Manual	All
☑	NCU access	Manual	All
☑	Notes about GSP	Manual	All

NCU access

Dialog	
Activation	Yes
Editing mode	Manual
Display messages	All
Operating instructions	
Text	**Hier können zusätzliche Hinweise für die Dialogseite durch d**
Address	
Preselection	**192.168.214.1**
NCU logon information	
User name	**manufact**
Password	●●●●●
Data backup to CF card	
Execute backup	No
Setting disabled	No
Archive	
Execute backup	No
Dialog scripts	
Scripts	**No script available**

Preselection
Default of the IP address of the NCU. This is added to the selection list in the package, and selected. IP 192.168.214.1/192.168.215.1 is always available in the selection list of the package for connections via X120/X127.

Figure 4-13 NCU access - project engineer view

"Display all system variables via a dialog" button

You open the "NCU access - System variables" dialog via the button (display all system variables via a dialog). All system variables to be used under "NCU access" are listed here.

Information on the system variables·can be found in the System variables for the NCU access dialog (Page 275) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

Address

The Address of the NCU must be entered for a remote Windows access from a PCU 50 or a PG/PC.

The "Preselection" is preset with the default address 192.168.214.1 during configuration.

Note

If the package is to be executed locally on an NCU, the specified address has no significance.

NCU login information

The following settings are available for the "NCU login information" property:

User name	Login information (user name, password) can be stored for access to the NCU. In a new package, these are preset to standard user of the NCU "manufact" and the appropriate password.
Password	

Data backup CF card

The following settings are available for the "Data backup CF card" property:

Perform backup	Yes	A data backup of the NCU can be selected on this dialog page at the start of the package execution. This backs up the content of the CF card, including the current active archive data (NC, PLC, DRV), in a folder named based on the CF card ID. If when starting, a data backup is found next to the package, it can be selected 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 an instruction *.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 CF card ID is to be restored (see section Creating MyConfig packages (Page 246)).
	No	
Setting disabled	Yes	The settings made during the configuration can only be viewed by the operator during execution, but not changed.
	No	

Note

for "Setting disabled"

The setting for selecting the data backup can be disabled for the operating sequence.

The licenses of the controller 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 system are to be retained.

Archive

A complete archive can also be created for the diagnostics, with a setting for all data to be backed up.

The complete archive is stored in a folder named based on the CF card ID with the name "Backup_<Date>.arc".

Dialog scripts

Under "Dialog scripts", entry "Scripts", you open the "ScriptEditor dialog - NCU access (<time of execution>)" dialog page via the ... button.
You can edit scripts here that are to be executed at selectable times during the package execution.

In this dialog, the specified dialog scripts of selected dialogs can be assigned to the desired events (=time) without having to call each individual dialog.

Assignment dialog - Event

The dialogs in which dialog scripts can be assigned to selected times (events), are listed in the Assignment dialog - Event (Page 101) table.

Meaning of the events:

Possible events and their meaning can be found in the Event - Meaning (Page 102) table.

Inserting a script

Information on inserting a script and on the following options can be found in the Scripting (Page 197) section at:

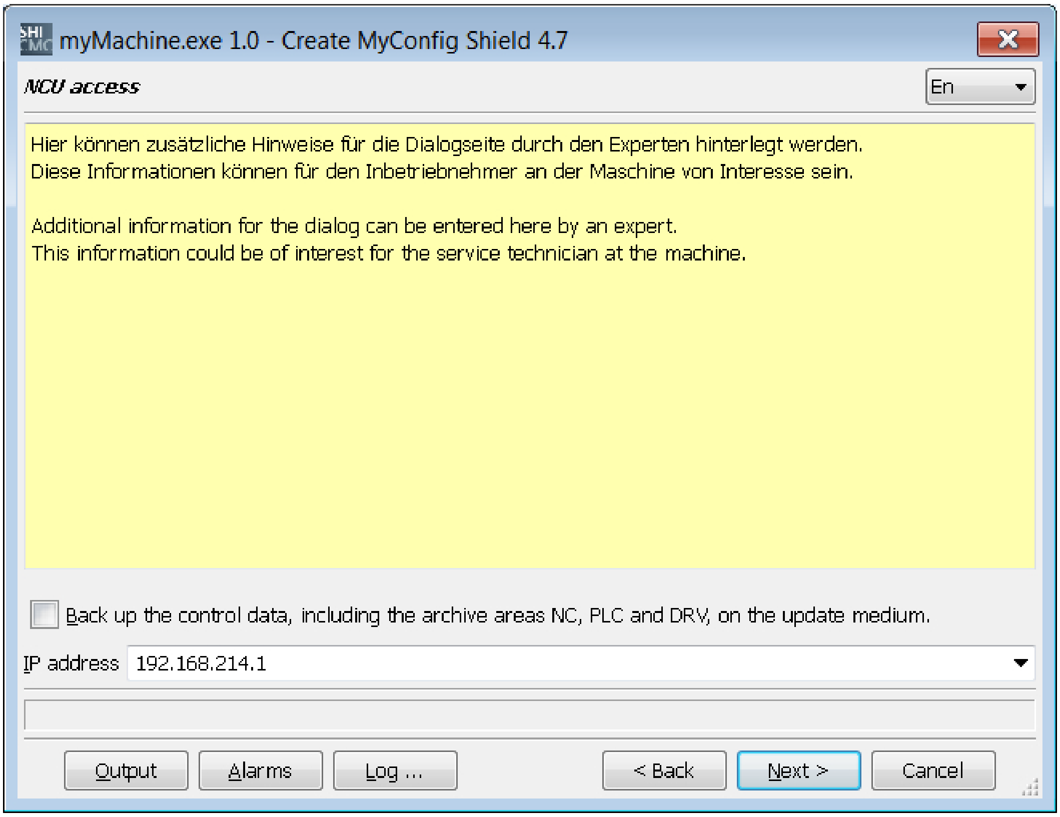
- Sections (Page 204)
- Area data (Page 206)
- Control structures (Page 206)
- Operators (Page 209)
- Functions (Page 212)
- Procedures (Page 231)

Dialog view during the package execution (for commissioning engineers / operators)

Execution of the "NCU access" dialog page cannot be deactivated.

If at "Package > Data areas > Use NCU" has been selected in the Project / Package tab, this dialog page is executed automatically when executing a package.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



Hier können zusätzliche Hinweise für die Dialogseite durch den Experten hinterlegt werden.
Diese Informationen können für den Inbetriebnehmer an der Maschine von Interesse sein.

Additional information for the dialog can be entered here by an expert.
This information could be of interest for the service technician at the machine.

□ Back up the control data, including the archive areas NC, PLC and DRV, on the update medium.

IP address 192.168.214.1

Figure 4-14 NCU access dialog - commissioning engineer / operator view

Buttons in dialogs

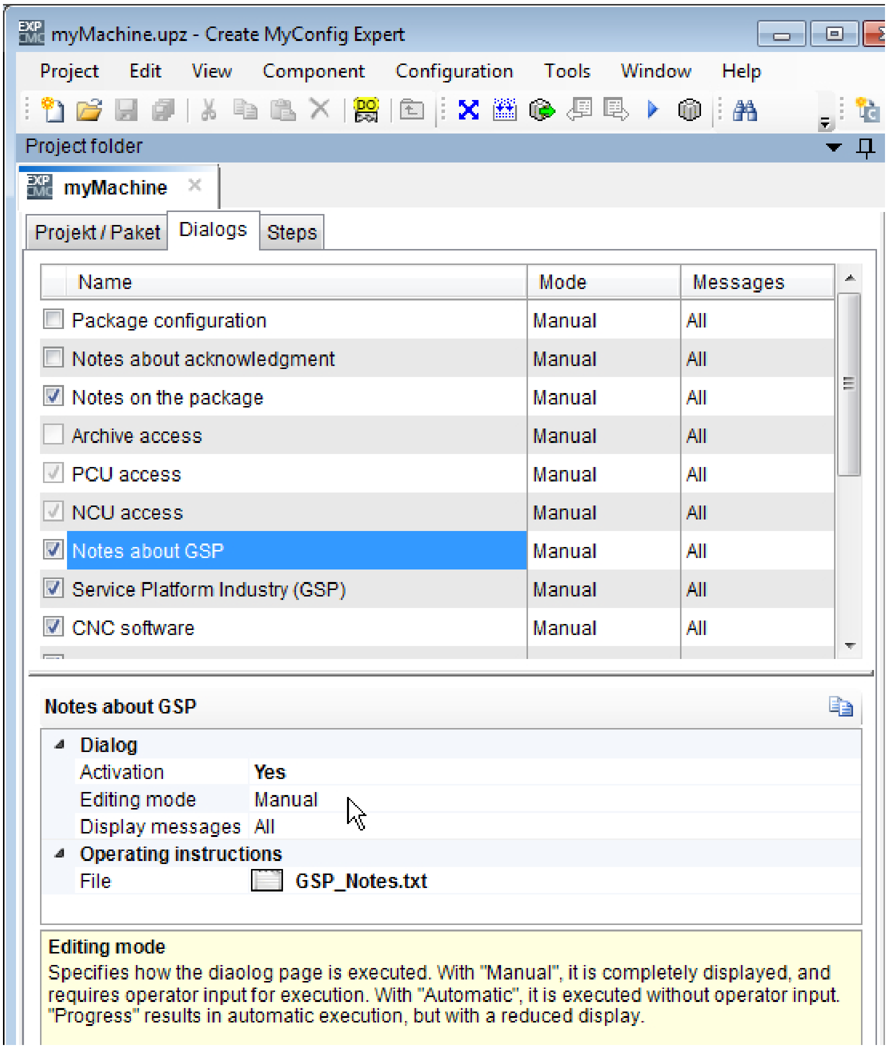
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.7 Dialog page: Notes on the Service Platform Industry (GSP)

Configuration

You can configure the following property groups for the "Notes on the Service Platform Industry (GSP)" dialog page:

- General
- Operating instructions



	Name	Mode	Messages
□	Package configuration	Manual	All
□	Notes about acknowledgment	Manual	All
☑	Notes on the package	Manual	All
□	Archive access	Manual	All
☑	PCU access	Manual	All
☑	NCU access	Manual	All
☑	Notes about GSP	Manual	All
☑	Service Platform Industry (GSP)	Manual	All
☑	CNC software	Manual	All

Notes about GSP

Dialog	
Activation	**Yes**
Editing mode	Manual
Display messages	All
Operating instructions	
File	**GSP_Notes.txt**

Editing mode
Specifies how the diaolog page is executed. With "Manual", it is completely displayed, and requires operator input for execution. With "Automatic", it is executed without operator input. "Progress" results in automatic execution, but with a reduced display.

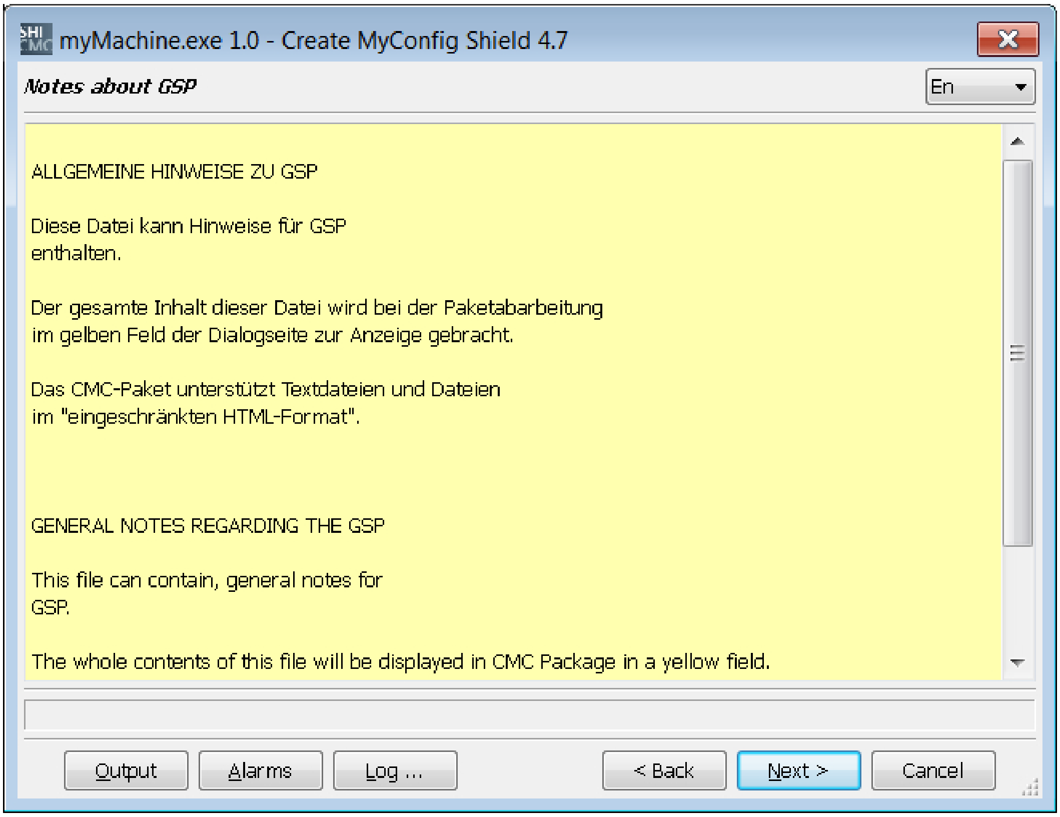
Figure 4-15 Notes on the Service Platform Industry (GSP) - Configuration engineer view

General / Operating instructions

More detailed information about the property groups "General" and "Operating instructions" can be found under Dialog page: Notes about acknowledgment (Page 104).

Package execution dialog

A file in text format or restricted HTML format with operating instructions for the following dialog can be displayed on the "Notes on Service Platform Industry (GSP)" dialog.



ALLGEMEINE HINWEISE ZU GSP

Diese Datei kann Hinweise für GSP
enthalten.

Der gesamte Inhalt dieser Datei wird bei der Paketabarbeitung
im gelben Feld der Dialogseite zur Anzeige gebracht.

Das CMC-Paket unterstützt Textdateien und Dateien
im "eingeschränkten HTML-Format".

GENERAL NOTES REGARDING THE GSP

This file can contain, general notes for
GSP.

The whole contents of this file will be displayed in CMC Package in a yellow field.

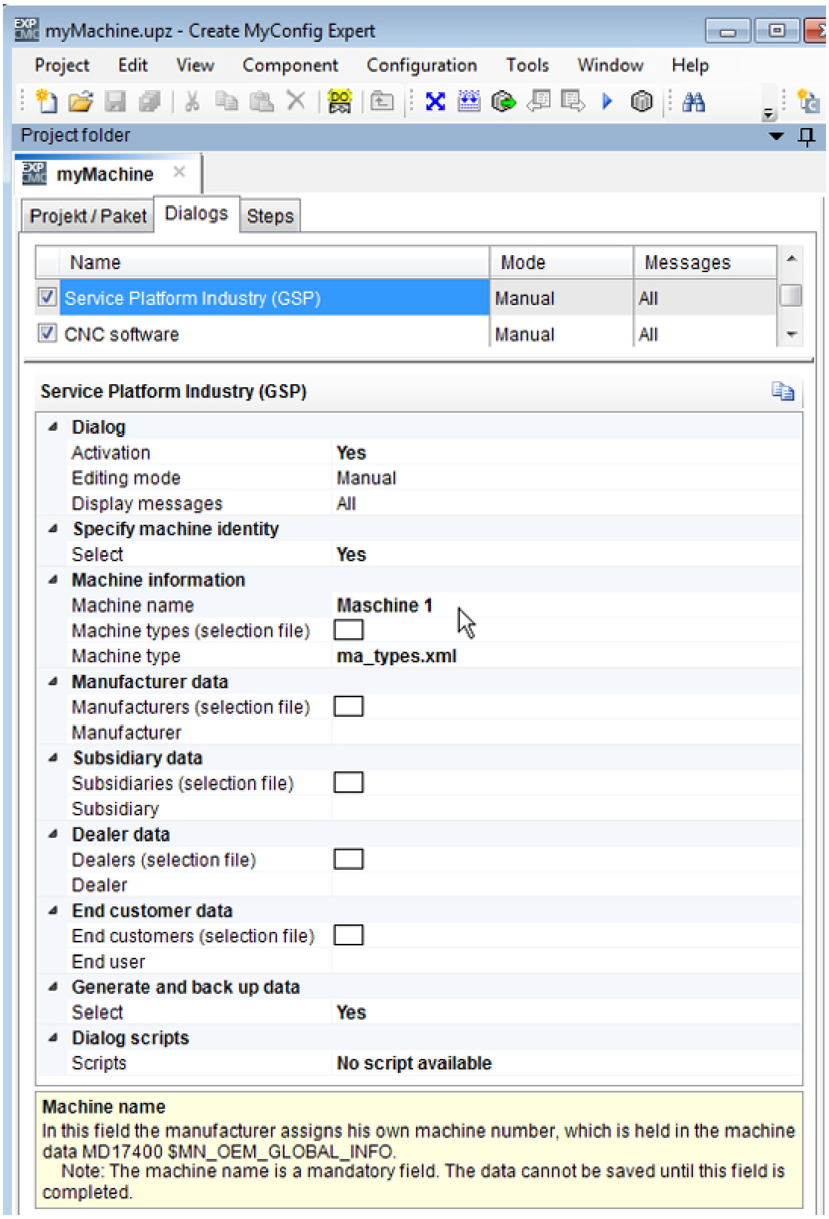
Figure 4-16 Notes on the Service Platform Industry (GSP) - commissioning engineer/operator view

4.3.2.8 Dialog page: Service Platform Industry (GSP)

Configuration

You can configure the following property groups for the "Service Platform Industry (GSP)" dialog page:

- Dialog
- Specify machine identity
- Machine information
- Manufacturer data
- Subsidiary data
- Dealer data
- End customer data
- Generate and back up data
- Dialog scripts



	Name	Mode	Messages
☑	Service Platform Industry (GSP)	Manual	All
☑	CNC software	Manual	All

Service Platform Industry (GSP)

Dialog	
Activation	**Yes**
Editing mode	Manual
Display messages	All
Specify machine identity	
Select	**Yes**
Machine information	
Machine name	**Maschine 1**
Machine types (selection file)	□
Machine type	**ma_types.xml**
Manufacturer data	
Manufacturers (selection file)	□
Manufacturer	
Subsidiary data	
Subsidiaries (selection file)	□
Subsidiary	
Dealer data	
Dealers (selection file)	□
Dealer	
End customer data	
End customers (selection file)	□
End user	
Generate and back up data	
Select	**Yes**
Dialog scripts	
Scripts	**No script available**

Machine name
In this field the manufacturer assigns his own machine number, which is held in the machine data MD17400 $MN_OEM_GLOBAL_INFO.
Note: The machine name is a mandatory field. The data cannot be saved until this field is completed.

Figure 4-17 Service Platform Industry (GSP) - project engineer view

Display all system variables via a dialog button

You open the "Service Platform Industry (GSP) - System variables" dialog via the button (display all system variables via a dialog). All system variables to be used under "Service Platform Industry (GSP)" are listed here.

Information on the system variables·can be found in the System variables for the Service Platform Industry (GSP) dialog (Page 280) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog

More detailed information about the property group "Dialog" can be found under the dialog page: Notes about acknowledgement (Page 104).

Specify machine identity

The following settings are available for the property "Specify machine identity":

Selection	Yes	Here you specify whether machine identity data is generated and saved for the GSP data.
	No	

Machine information

The following settings are available for the "Machine information" property:

Machine name	Here, you specify the name of the machine. **Notice** The machine name is only imported if the "NC archive data (Page 131)" is used on the "System configuration" dialog page.
Machine types (selection file)	The "ma_types.xml" file that contains a list of machine types can be linked or inserted here.
Machine type	Click … to display the list of contained machine types and to select a machine type.

Example of file "ma_types.xml" (source: DOC_ON_CD):

```
<?xml version="1.0" encoding="utf-8"?>
<Machine types>
<Type>LC 80</Type>
<Type>LC 82</Type>
<Type>LC 120</Type>
<Type>LC 122</Type>
<type>LC 150</type>
</Machine types>
```

Manufacturer data

The following settings are available for the "Manufacturer data" property:

Manufacturers (selection file)	The "oem.xml" file that contains a list of manufacturers with their addresses can be linked or inserted here.
Manufacturer	The manufacturers contained in the selection file are displayed via the context menu and can be selected.

Subsidiary data

The following settings are available for the "Subsidiary data" property:

Subsidiary (selection file)	The "oemsubs.xml" file that contains a list of branch offices with their addresses can be linked or inserted here.
Subsidiaries	The subsidiaries contained in the selection file are displayed via the context menu and can be selected.

Dealer data

The following settings are available for the "Dealer data" property:

Dealer (selection file)	The "dealer.xml" file that contains a list of dealers with their addresses can be linked or inserted here.
Dealer	The dealers contained in the selection file are displayed via the context menu and can be selected.

Example of file "dealer.xml" (source: DOC_ON_CD)

You can acquire the address data of any number of dealers in file "dealer.xml". If you wish to enter additional dealers, copy the <Dealer> to </Dealer> area for each dealer.

```
<?xml version="1.0" encoding="utf-8"?>
<Addresses>
<Dealer>
<Customer-Id></Customer-Id>
<Name></Name>
<Street></Street>
<ZIP code></ZIP code>
<Location></Location>
<Country></Country>
<State></State>
<Contact></Contact>
<Phone></Phone>
<Fax></Fax>
<E-mail></E-mail>
<URL></URL>
</Dealer>
</Addresses>
```

End customer data

The following settings are available for the "End customer data" property:

End customers (selection file)	The "user.xml" file that contains a list of end customers with their addresses can be linked or inserted here.
End customer	The end customers contained in the selection file are displayed via the context menu and can be selected.

Generate and back up data

The following settings are available for the "Generate and back up data" property:

Selection	Yes	Here, you specify whether the **GSP data** is to be generated after executing the step and in stored in a folder named based on the CF card ID.
	No	

Dialog scripts

Under "Dialog scripts", entry"Scripts", the ... button calls the "ScriptEditor dialog - Service Platform Industry (GSP) (<Time of the execution>)" dialog page.
You can edit scripts here that are to be executed at selectable times during the package execution.

In this dialog, the specified dialog scripts of selected dialogs can be assigned to the desired events (=time) without having to call each individual dialog.

Assignment dialog - Event

The dialogs in which dialog scripts can be assigned to selected times (events), are listed in the Assignment dialog - Event (Page 101) table.

Meaning of the events:

Possible events and their meaning can be found in the Event - Meaning (Page 102) table.

Inserting a script

Information on inserting a script and on the following options can be found in the Scripting (Page 197) section at:

- Sections (Page 204)
- Area data (Page 206)
- Control structures (Page 206)
- Operators (Page 209)
- Functions (Page 212)
- Procedures (Page 231)

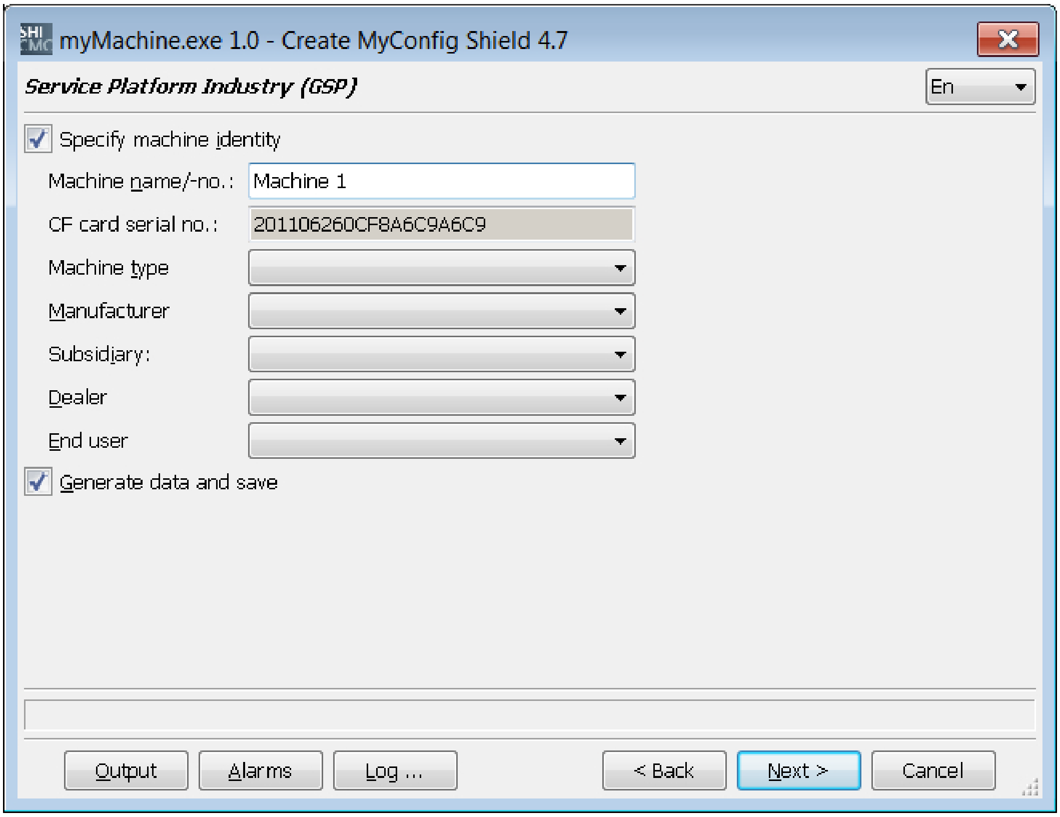
Dialog view during the package execution (for commissioning engineers / operators)

GSP data can be parameterized in this dialog.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).

Note

Templates for the described XML files can be viewed on the NCU under "/card/siemens/sinumerik/hmi/template/identity".



myMachine.exe 1.0 - Create MyConfig Shield 4.7

Service Platform Industry (GSP)

☑ Specify machine identity

Machine name/-no.:	Machine 1
CF card serial no.:	201106260CF8A6C9A6C9
Machine type	
Manufacturer	
Subsidiary:	
Dealer	
End user	

☑ Generate data and save

Figure 4-18 Service Platform Industry (GSP) - commissioning engineer / operator view

Buttons in dialogs

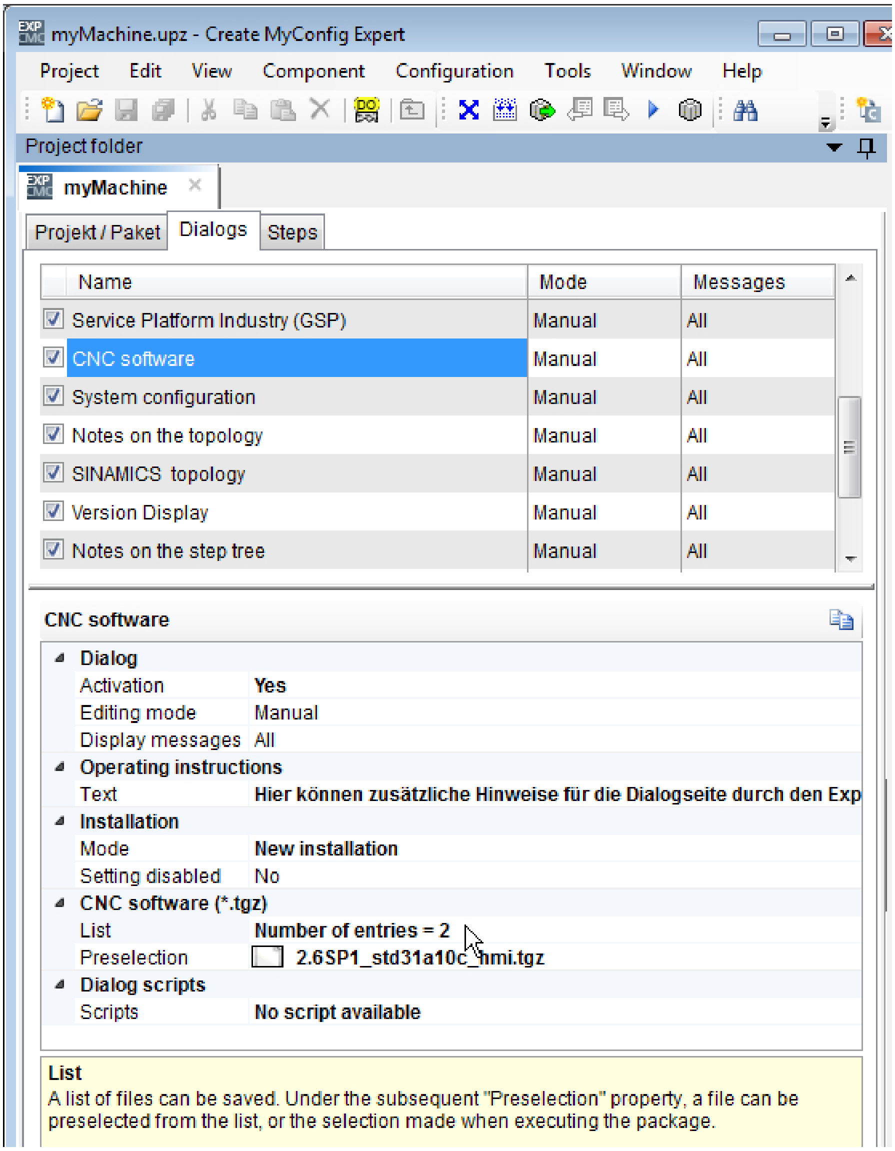
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.9 Dialog: CNC software

Configuration

You can configure the following property groups for the "CNC software" dialog page:

- Dialog
- Operating instructions
- Installation
- CNC software (*.tgz)
- Dialog scripts



	Name	Mode	Messages
☑	Service Platform Industry (GSP)	Manual	All
☑	CNC software	Manual	All
☑	System configuration	Manual	All
☑	Notes on the topology	Manual	All
☑	SINAMICS topology	Manual	All
☑	Version Display	Manual	All
☑	Notes on the step tree	Manual	All

CNC software

Dialog	
Activation	**Yes**
Editing mode	Manual
Display messages	All
Operating instructions	
Text	**Hier können zusätzliche Hinweise für die Dialogseite durch den Exp**
Installation	
Mode	**New installation**
Setting disabled	No
CNC software (*.tgz)	
List	**Number of entries = 2**
Preselection	□ **2.6SP1_std31a10c_hmi.tgz**
Dialog scripts	
Scripts	**No script available**

List
A list of files can be saved. Under the subsequent "Preselection" property, a file can be preselected from the list, or the selection made when executing the package.

Figure 4-19 CNC software - project engineer view

Display all system variables via a dialog button

You open the "CNC software - System variables" dialog via the button (display all system variables via a dialog). All system variables to be used under "CNC software" are listed here.

Information on the system variables·can be found in the section System variables for the CNC software dialog (Page 276) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

Installation

The following settings are available for the "Installation" property:

Mode	None	Here, you select the type of installation. "New installation" means that the CF card is deleted and the system software is reinstalled. The subsystems NC, PLC and DRV are overall reset and default data is loaded. In the "Upgrade" mode, the NC, PLC and DRV data is migrated immediately after installing the new CNC software, i.e. before the step is executed. **Note** Data migration means that the backed-up archive is imported from the old to the new installed software version. Step execution will be performed later based on the archive data of the newly installed software version.
	New installation	
	Upgrade	
Setting disabled	Yes	Setting for the mode can be disabled for the operation sequence.
	No	

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 and can be inserted as copy or link in a list. Linked files are not taken into this list until the package is deployed.
Preselection	A file can be preselected from the list. If no file is preselected, the operator must make the selection on the machine. With the specification of './<Path>/<File>', a file next to the package can be preselected. For example, ./std31a10c_hmi.tgz is then in the preselection.

Note

If the package is started from a service system USB FlashDrive, then generally booting is from the CF card after this dialog.

Dialog scripts

Under "Dialog scripts", entry"Scripts", the ... button calls the "ScriptEditor dialog - CNC software (*.tgz) (<Time of the execution>)" dialog page.
You can edit scripts here that are to be executed at selectable times during the package execution.

In this dialog, the specified dialog scripts of selected dialogs can be assigned to the desired events (=time) without having to call each individual dialog.

Assignment dialog - Event

The dialogs in which dialog scripts can be assigned to selected times (events), are listed in the Assignment dialog - Event (Page 101) table.

Meaning of the events:

Possible events and their meaning can be found in the Event - Meaning (Page 102) table.

Inserting a script

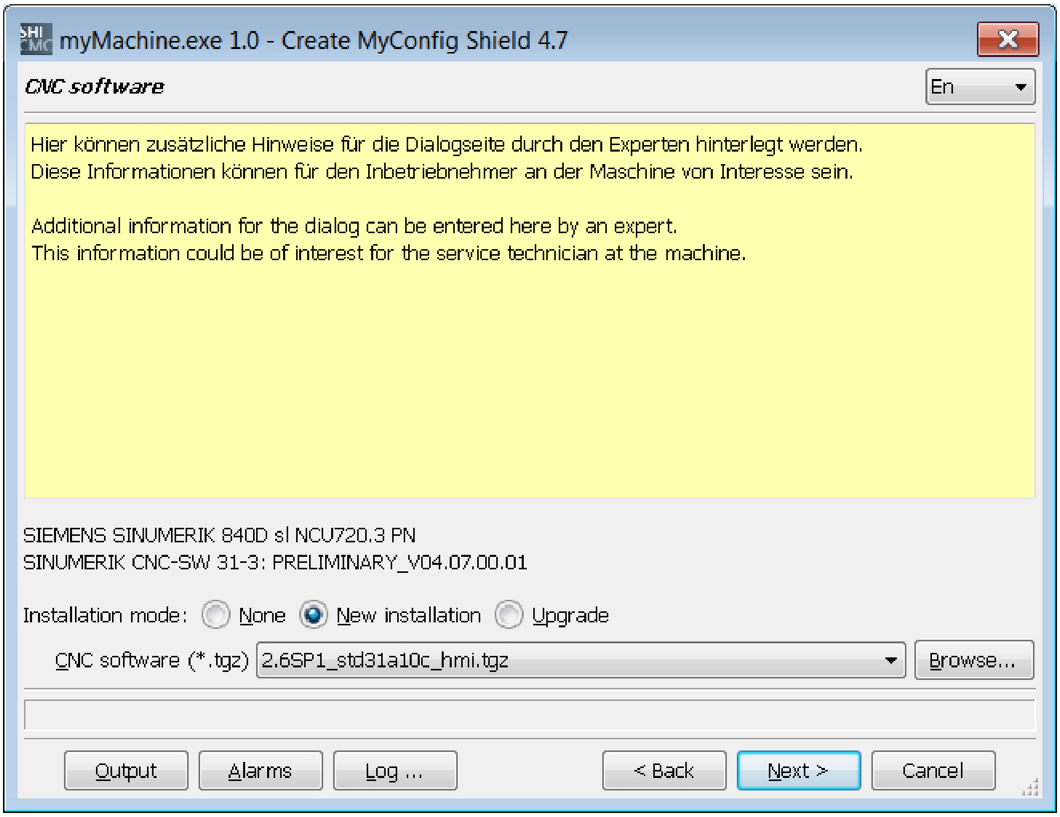
Information on inserting a script and on the following options can be found in the Scripting (Page 197) section under:

- Sections (Page 204)
- Area data (Page 206)
- Check structures (Page 206)
- Operators (Page 209)
- Functions (Page 212)
- Procedures (Page 231)

Dialog view during the package execution (for commissioning engineers / operators)

The "CNC software" dialog box is used for installation of new CNC software or for upgrading existing CNC software.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



Hier können zusätzliche Hinweise für die Dialogseite durch den Experten hinterlegt werden.
Diese Informationen können für den Inbetriebnehmer an der Maschine von Interesse sein.

Additional information for the dialog can be entered here by an expert.
This information could be of interest for the service technician at the machine.

SIEMENS SINUMERIK 840D sl NCU720.3 PN
SINUMERIK CNC-SW 31-3: PRELIMINARY_V04.07.00.01

Installation mode: ○ None ● New installation ○ Upgrade

CNC software (*.tgz) 2.6SP1_std31a10c_hmi.tgz

Figure 4-20 CNC software dialog - commissioning engineer / operator view

Using the "CNC software (*.tgz)" drop-down list, a file that was assigned to the package or a file lying next to the package (prefix: "./") can be selected.
The "Browse" button allows a selection to be made from any location.

After the TGZ file has been loaded, the NCU is automatically restarted.

Note

TGZ files generated using "sc backup xyz.tgz" or backups generated using Create MyConfig can **only** be used for a new installation.

They **cannot** be used for an upgrade.

If a backup of CMC is used for a new installation, the archive data it contains will be imported.

Buttons in dialogs

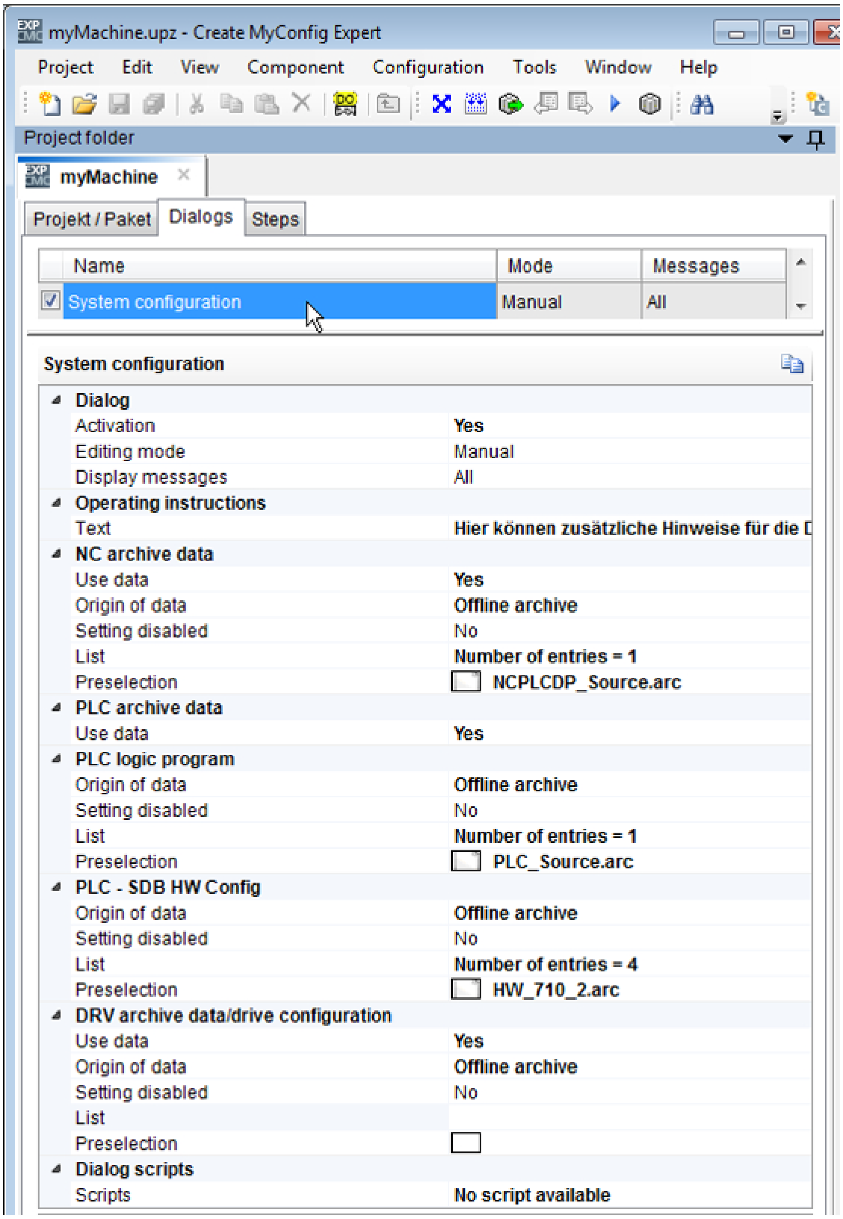
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.10 Dialog: System configuration

Configuration

You can configure the following property groups for the "System configuration" dialog page:

- Dialog
- Operating instructions
- NC archive data
- PLC archive data
- PLC logic program
- PLC - SDB HW Config
- DRV archive data / drive configuration
- Dialog scripts



	Name	Mode	Messages
☑	System configuration	Manual	All

System configuration

Dialog	
Activation	**Yes**
Editing mode	Manual
Display messages	All
Operating instructions	
Text	**Hier können zusätzliche Hinweise für die D**
NC archive data	
Use data	**Yes**
Origin of data	**Offline archive**
Setting disabled	No
List	**Number of entries = 1**
Preselection	□ **NCPLCDP_Source.arc**
PLC archive data	
Use data	**Yes**
PLC logic program	
Origin of data	**Offline archive**
Setting disabled	No
List	**Number of entries = 1**
Preselection	□ **PLC_Source.arc**
PLC - SDB HW Config	
Origin of data	**Offline archive**
Setting disabled	No
List	**Number of entries = 4**
Preselection	□ **HW_710_2.arc**
DRV archive data/drive configuration	
Use data	**Yes**
Origin of data	**Offline archive**
Setting disabled	No
List	
Preselection	□
Dialog scripts	
Scripts	**No script available**

Figure 4-21 System configuration - project engineer view

The source or output data is provided via the "System configuration" dialog page.
This data is then manipulated later on the "SINAMICS topology" and "Step tree" dialog pages.
At the end of the execution of the "Step tree" dialog page, the changed data is transferred to the control or the result archive.

On the "System configuration" dialog page, you can define for every archive area (NC, PLC, DRV) whether it should be used.

For "Source of the data", select from where the data of the individual areas should be sourced. This selection can be blocked individually for each area by the project engineer in order to prevent changes by the operator during the package execution.

The PLC area is subdivided into the PLC logic and PLC SDB.

Display all system variables via a dialog button

You open the "System configuration - System variables" dialog via the button 📋 (display all system variables via a dialog). All system variables to be used under "System configuration" are listed here.

Information on the system variables·can be found in the System variables for the System configuration dialog (Page 278) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

NC archive data

The following settings are available for the "NC archive data" and are activated when the package is executed:

Table 4-10 NC archive data

Use data	Yes	Here, you specify whether the "NC archive data" is to be used.
	No	
Origin of data	**Note** This selection can be blocked by the project engineer in order to prevent changes by the operator during the package execution.	
	Initial state	If the "NCU area" was activated during package configuration, the archive data can be loaded directly from the control with the setting "Initial state". If the initial state is used, the archive is extracted that was located on the control when this dialog box was entered. If the CNC software was installed beforehand, only default data is available in the archive. **Note** No data is reset when upgrading the CNC software. The archive that is available in the system after data migration is used.
	Overall reset	If this option is used, default data is loaded and then the archive data is backed up.
	Offline archive	If this option is used, the output data for the NC area is taken from a stored SINUMERIK archive.
Setting disabled	No	The selection at "Source of the data" can be blocked by the project engineer via this option in order to prevent changes by the operator during the package execution.
	Yes	

List	Under "List", SINUMERIK archives with NC data can be inserted or linked for "Offline Archive". These archives are also saved in the package during deployment and can be selected during package execution. Several archives can be linked or inserted via the "List" property, which is available for the preselection or during the package execution for the selection.
Preselection	Under "Preselection", a SINUMERIK archive can be inserted, linked or preselected from the list for "Offline Archive". Select an archive from the "List", or enter the path of a file which is located next to the package "/<path>/<file>", e.g. "./NC-data.arc". **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:

Use data	Yes	Here you define whether the archive area "PLC archive data" with the subareas "PLC logic program" and "PLC - SDB HW Config" are used in the package during configuration. A change is possible via the package configuration or dialog script.
	No	

PLC logic program

The following settings are available for "PLC - Logic program" 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 the package execution.	
	Initial state	If the "NCU area" was activated during package configuration, the archive data can be loaded directly from the control with the setting "Initial state". **Note** No data is reset when upgrading the CNC software. The archive that is available in the system after data migration is used.
	Overall reset	If this option is used, the PLC memory is overall reset and then the archive data is backed up.
	Offline archive	If this option is used, the output data for the PLC area is taken from a stored SINUMERIK archive.
Setting disabled	No	The selection at "Source of the data" can be blocked by the project engineer via this option in order to prevent changes by the operator during the package execution.
	Yes	

List	Under "List", SINUMERIK archives with PLC data can be inserted or linked for "Offline Archive". These archives are also saved in the package during deployment and can be selected during package execution. Using the "List" property, multiple archives can be transferred, one of which must be selected at runtime.
Preselection	Under "Preselection", a SINUMERIK archive with PLC data (blocks) can be inserted, linked or preselected from the list for "Offline Archive". Alternatively, enter the path of a file which is located next to the package "/<Path>/<File>", e.g. "./sdb.arc". **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 the data" must be set to "Overall reset".

PLC - SDB HW Config

The following settings are available for "PLC - SDB HW Config" 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 the package execution.	
	Initial state	If the "NCU area" was activated during package configuration, the archive data can be loaded directly from the control with the setting "Initial state". If the initial state is used, the archive is extracted that was located on the control when this dialog box was entered. If the CNC software was installed beforehand, only default data is available in the archive. **Note** No data is reset when upgrading the CNC software. The archive that is available in the system after data migration is used.
	Overall reset	When this option is used, the PLC system data memory is overall reset and then the archive data is backed up. **NOTICE** Drive configuration is not possible on a PLC-SDB HW configuration on which an overall memory reset has been performed.
	Offline archive	If this option is used, the output data for the area is taken from a stored SINUMERIK archive.
Setting disabled	No	The selection at "Source of the data" can be blocked by the project engineer via this option in order to prevent changes by the operator during the package execution.
	Yes	

List	Under "List", SDB archives with system data blocks can be inserted or linked for "Offline Archive". These archives are also saved in the package during deployment and one of them can be selected during package execution. Several archives can be linked or inserted via the "List" property, which is available for the preselection or during the package execution for the selection.
Preselection	Under "Preselection", an SDB archive with system data blocks can be inserted, linked or preselected from the list for "Offline Archive". The package specified here is preselected during the package execution. An SDB archive can also be saved in addition to the package when the package is being executed. Alternatively, enter the path of a file which is located next to the package "/<Path>/<File>", e.g. "./sdb.arc".

DRV archive data / drive configuration

In the "DRV archive data / drive configuration" property group, you can define via the "Source of the data" property whether drive commissioning will be performed in the production step by Automatic configuration or User-specified topology.

The following settings are available for the "DRV archive data / drive configuration" and are activated when the package is executed:

Use data	Yes	Here, you define whether the archive area "DRV archive data" is used in the package during configuration. A change is possible via the package configuration or dialog script.
	No	

Origin of data	Initial state	If the "NCU area" was activated during package configuration, the archive data can be loaded directly from the control with the setting "Initial state". **Note** No data is reset when upgrading the CNC software. The archive that is available in the system after data migration is used. During a re-installation of the CNC software, a blank 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 (incl. DO sorting, BICO interconnection, etc...) is performed. In this mode, you can work with a "comparison topology (*.ust)", which must be inserted in the "SINAMICS topology" dialog page. This means that the automatically identified SINAMICS target topology can be compared with a comparison topology generated offline. In "Automatic configuration", SINAMICS determines the target topology, which determines all modules connected at this time, including their DRIVE-CLiQ wiring. In the subsequent package stages, the most recently created online archive is always used. It is only possible to modify the topology by adding, removing, or replacing components via SINUMERIK Operate or configuration must be performed again.
	User-specified topology	With the selection "User-specified topology", a SINAMICS offline topology package (*.utz) is loaded during package execution. For drive commissioning, a "user-specified topology" can be loaded in a SINAMICS offline topology package that also contains modules that will not be attached, wired, and activated until later during commissioning. The necessary data objects are created by SINAMICS also for the modules that are not yet present, too, so that the user data can already be loaded during package execution. In the subsequent package stages, the most recently created online archive is always used. A main application case for this are DQI/SMI motors. The Cu and the NX expansion modules must always be present. **Note** When selecting the option "User-specified topology option, the appropriate user-specified topology (*.utz) must also be entered in the "SINAMICS topology" dialog.
	Offline archive	When the "Offline Archive" is set, the data from a SINUMERIK archive is used.
Setting disabled	Yes	The selection at "Source of the data" can be blocked by the project engineer via this option in order to prevent changes by the operator during the package execution.
	No	

List	Under "List", SINUMERIK archives with drive data can be inserted or linked for the "Offline archive". These archives are also saved in the package during deployment and can be selected during package execution. Several archives can be linked or inserted via the "List" property, which is available for the preselection or during the package execution for the selection.
Preselection	Under "Preselection", a SINUMERIK archive can be inserted, linked or preselected from the list for "Offline Archive". Alternatively, enter the path of a file which is located next to the package "/<path>/<file>", e.g. "./DRV-data.arc". **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 the data" must be set to "Overall reset".

Dialog scripts

Under "Dialog scripts", entry"Scripts", the ... button calls the "ScriptEditor dialog - System configuration (<Time of the execution>)" dialog page.
You can edit scripts here that are to be executed at selectable times during the package execution.

In this dialog, the specified dialog scripts of selected dialogs can be assigned to the desired events (=time) without having to call each individual dialog.

Assignment dialog - Event

The dialogs in which dialog scripts can be assigned to selected times (events), are listed in the Assignment dialog - Event (Page 101) table.

Meaning of the events:

Possible events and their meaning can be found in the Event - Meaning (Page 102) table.

Inserting a script

Information on inserting a script and on the following options can be found in the Scripting (Page 197) section under:

- Sections (Page 204)
- Area data (Page 206)
- Check structures (Page 206)
- Operators (Page 209)
- Functions (Page 212)
- Procedures (Page 231)

Package execution

If the "System configuration" dialog page is executed in editing mode "Manual", the "Origin of data" can be adapted for every area if the project engineer has not locked this.

In the case of "Offline archive", archives must be selected that were either transferred with the package, are located next to the package (prefix "./") or are inserted at any location using the ... ("Search") button.

To address offline archives from the control or from the archive path, the following prefixes are provided:

- **ARC**: Archive path that was specified under "Windows access"
- **oem**: Operate oem/sinumerik/data/archive
- **user**: Operate user/sinumerik/data/archive
- **ADV**: HMI-Adv F:\DH\ARC.DIR

Note

If, during package execution, a jump is made to the System configuration dialog page via "Back", then the archives backed up there are reused, and changes that have possibly already been made are rejected.

Examples

Example: Archive next to the package:

./myPath/myArchive.arc
An archive "myArchive.arc" from the folder "myPath", which is located next to the package, is used.
The prefix "./" stands for the current path in which the package file is located at the time of execution.

Example: Archive path as under "Access Windows":

If under "Access Windows", the path "D:\PCU50" has been specified and an archive "myArchive.arc" exists under this path, then this archive can be selected with the entry "ARC:myArchive.arc".

Example: Archive selection of PCU50 Windows 7 (USER)

If an archive "myArchive.arc" exists on PCU50 V5 in the path "C:\Program Files (x86)\Siemens\MotionControl\user\sinumerik\data\archive\", then this archive can be selected in the selection with the entry "user:myArchive.arc".

Example: Archive selection of PCU50 Windows 7 (OEM)

If an archive "myArchive.arc" exists on PCU50 V5 in the path "C:\Program Files (x86)\Siemens\MotionControl\oem\sinumerik\data\archive\", then this archive can be selected in the options with the entry "oem:myArchive.arc".

Example: Archive selection of PCU50 HMI Advanced

If an archive exists on PCU50 with HMI Advanced, it was saved under "F:\dh\arc.dir".
This archive can be selected in the options, e.g. with the entry "ADV:MYARCHIV.ARC (MYARCHIVE.ARC)".

Dialog view during the package execution (for commissioning engineers / operators)

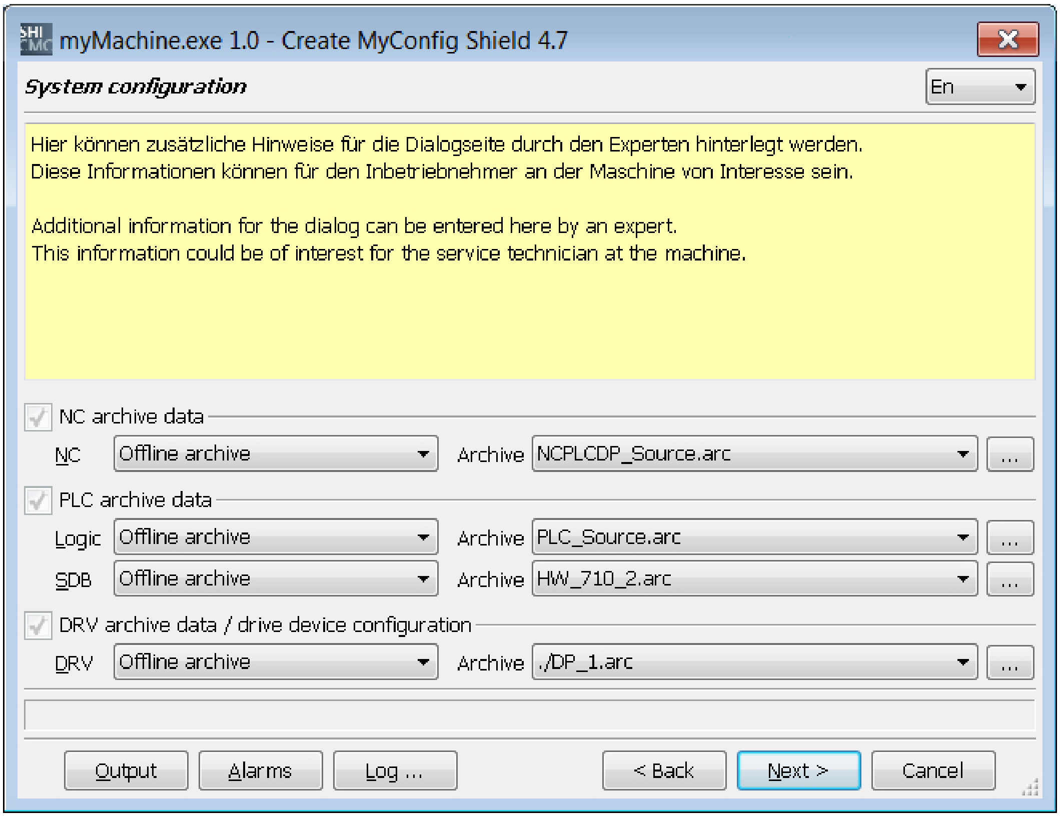
In this dialog you set which archive data (NC, PLC, and DRV) will be used later in the subsequent package stages.

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, it is necessary to import an appropriate PLC hardware configuration in the form of an SDB archive. It is not absolutely necessary that the final hardware expansion is included in this SDB archive, but only the information valid for this NCU relating to PROFIBUS, PROFINET, NCU, and NX.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



myMachine.exe 1.0 - Create MyConfig Shield 4.7

System configuration

Hier können zusätzliche Hinweise für die Dialogseite durch den Experten hinterlegt werden.
Diese Informationen können für den Inbetriebnehmer an der Maschine von Interesse sein.

Additional information for the dialog can be entered here by an expert.
This information could be of interest for the service technician at the machine.

☑ NC archive data
NC: Offline archive — Archive: NCPLCDP_Source.arc

☑ PLC archive data
Logic: Offline archive — Archive: PLC_Source.arc
SDB: Offline archive — Archive: HW_710_2.arc

☑ DRV archive data / drive device configuration
DRV: Offline archive — Archive: ./DP_1.arc

Figure 4-22 System configuration dialog - commissioning engineer / operator view

Buttons in dialogs

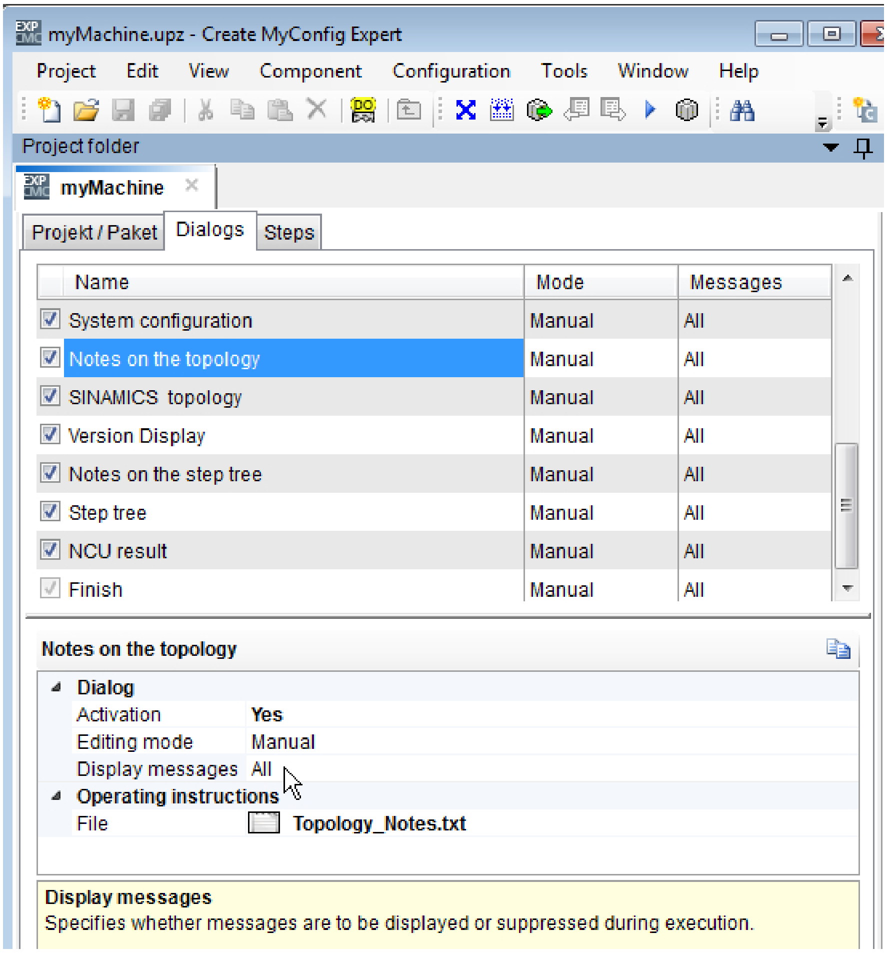
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.11 Dialog: Notes on the topology

Configuration

You can configure the following property groups for the "Notes on the topology" dialog page:

- Dialog
- Operating instructions



Name	Mode	Messages
☑ System configuration	Manual	All
☑ Notes on the topology	Manual	All
☑ SINAMICS topology	Manual	All
☑ Version Display	Manual	All
☑ Notes on the step tree	Manual	All
☑ Step tree	Manual	All
☑ NCU result	Manual	All
☑ Finish	Manual	All

Notes on the topology

Dialog	
Activation	Yes
Editing mode	Manual
Display messages	All
Operating instructions	
File	Topology_Notes.txt

Display messages
Specifies whether messages are to be displayed or suppressed during execution.

Figure 4-23 Notes on the topology - project engineer view

Display all system variables via a dialog button

You open the "Notes for the topology - System variables" dialog via the button (display all system variables via a dialog). All system variables to be used under "Notes on the topology" are listed here.

Information on the "Notes on the topology - System variables" can be found in section System variables for Up.$Dialog.DriveTopologyNotes (Page 278).

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

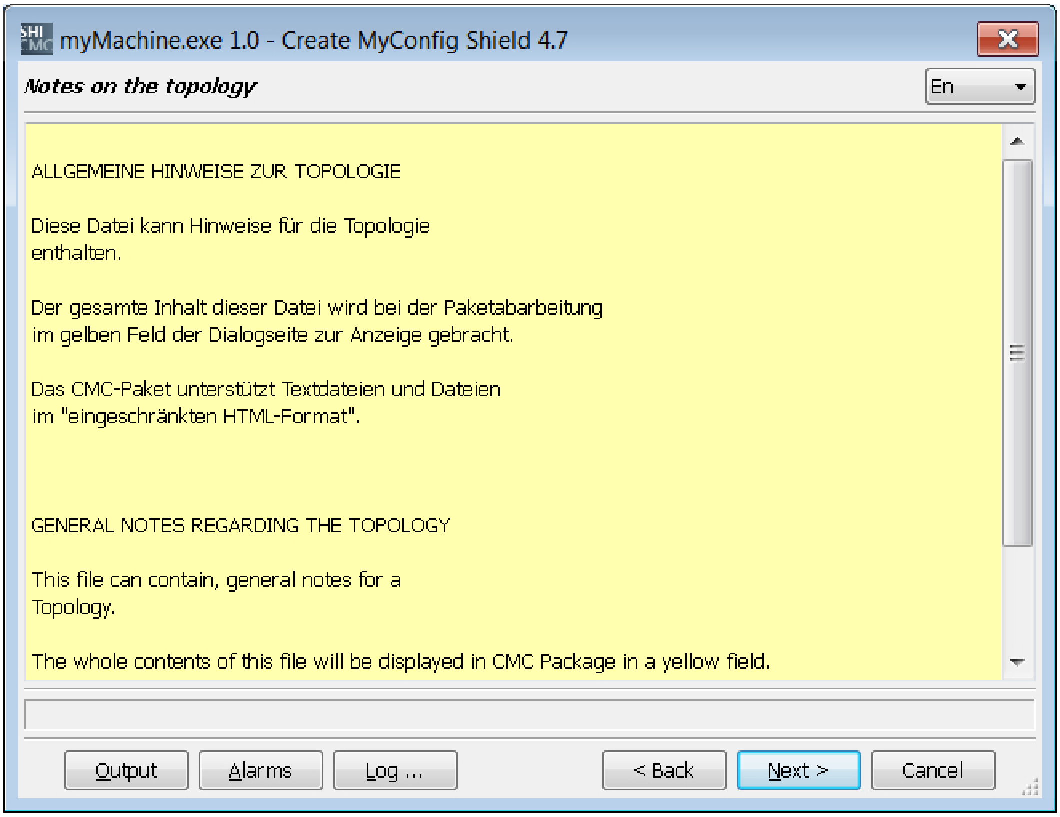
Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

Dialog view during the package execution (for commissioning engineers / operators)

A file in text format or restricted HTML format with operating instructions for the following dialog page can be displayed on the "Notes on the topology" dialog page.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



ALLGEMEINE HINWEISE ZUR TOPOLOGIE

Diese Datei kann Hinweise für die Topologie
enthalten.

Der gesamte Inhalt dieser Datei wird bei der Paketabarbeitung
im gelben Feld der Dialogseite zur Anzeige gebracht.

Das CMC-Paket unterstützt Textdateien und Dateien
im "eingeschränkten HTML-Format".

GENERAL NOTES REGARDING THE TOPOLOGY

This file can contain, general notes for a
Topology.

The whole contents of this file will be displayed in CMC Package in a yellow field.

Figure 4-24 Notes on the topology dialog - commissioning engineer / operator view

Buttons in dialogs

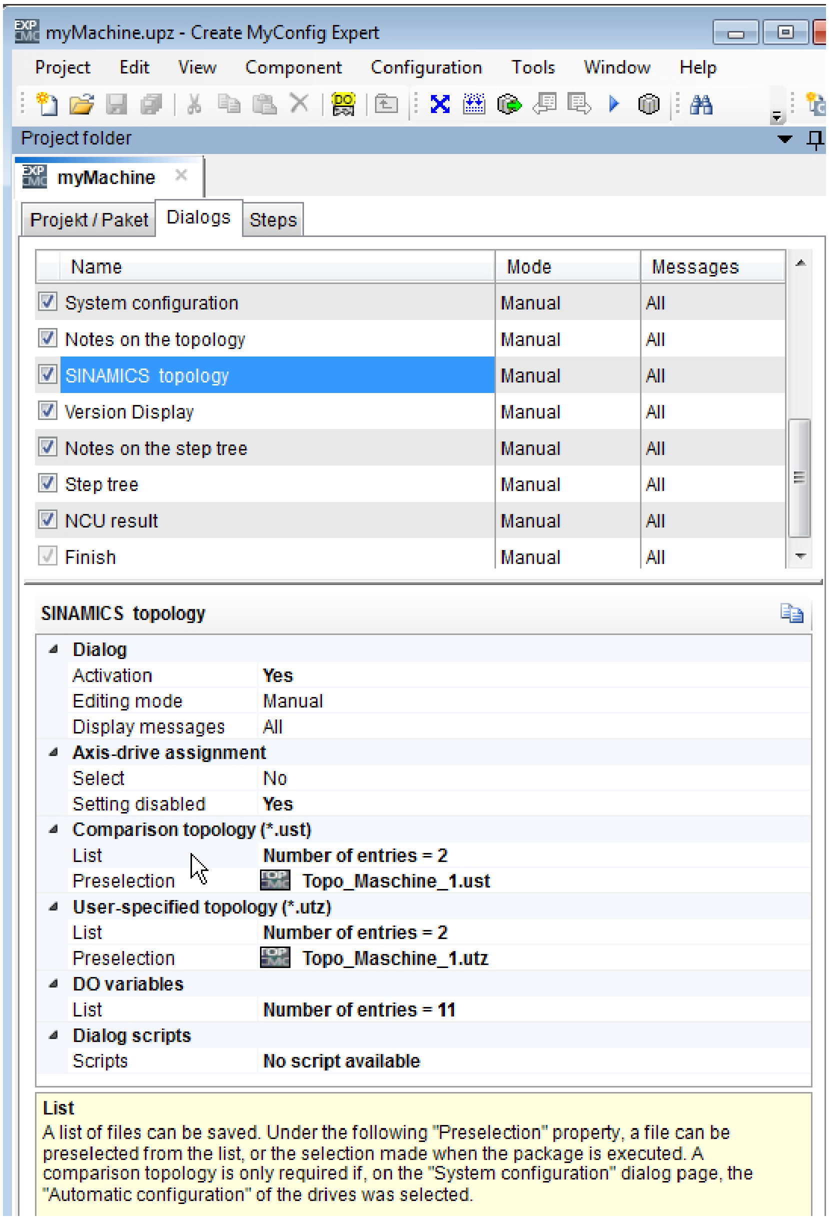
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.12 Dialog: SINAMICS topology

Configuration

You can configure the following property groups for the "SINAMICS topology" dialog page:

- Dialog
- Axis-drive assignment
- Comparison topology (*.ust)
- User-specified topology (*.utz)
- DO variables
- Dialog scripts



	Name	Mode	Messages
☑	System configuration	Manual	All
☑	Notes on the topology	Manual	All
☑	SINAMICS topology	Manual	All
☑	Version Display	Manual	All
☑	Notes on the step tree	Manual	All
☑	Step tree	Manual	All
☑	NCU result	Manual	All
☑	Finish	Manual	All

SINAMICS topology

Dialog	
Activation	**Yes**
Editing mode	Manual
Display messages	All
Axis-drive assignment	
Select	No
Setting disabled	**Yes**
Comparison topology (*.ust)	
List	**Number of entries = 2**
Preselection	**Topo_Maschine_1.ust**
User-specified topology (*.utz)	
List	**Number of entries = 2**
Preselection	**Topo_Maschine_1.utz**
DO variables	
List	**Number of entries = 11**
Dialog scripts	
Scripts	**No script available**

List
A list of files can be saved. Under the following "Preselection" property, a file can be preselected from the list, or the selection made when the package is executed. A comparison topology is only required if, on the "System configuration" dialog page, the "Automatic configuration" of the drives was selected.

Figure 4-25 SINAMICS topology - project engineer view

Display all system variables via a dialog button

You open the "SINAMICS topology - System variables" dialog via the button 📋 (display all system variables via a dialog). All system variables to be used under "SINAMICS topology" are listed here.

Information on the system variables·can be found in the System variables for the SINAMICS topology dialog (Page 279) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog

More detailed information about the property group "Dialog" can be found under the dialog page: Notes about acknowledgement (Page 104).

Axis-drive assignment

The following options are offered for the "axis-drive assignment", which are activated when the package is executed:

Selection	No	Here, you specify whether the "axis-drive assignment" is to be used.
	Yes	
Setting disabled	No	A change of the "Selection" property by the operator during the package execution can be blocked by the project engineer.
	Yes	

An axis-drive assignment can be performed in both variants, comparison topology and user-specified topology.

The checkbox must be activated to perform the assignment. The condition is that the selected topology contains an axis-drive assignment.

Note

For security reasons, Axis-drive assignment is only permissible with drive configuration "User-defined topology" and "Automatic configuration."

Comparison topology (*.ust)

Note

Prerequisite for comparison topology

On the "SINAMICS topology" dialog page, a comparison topology is always used if, on the System configuration dialog page under DRV archive data / Drive configuration, for "Origin of data" Initial state, Automatic configuration, or Offline archive has been selected.

With the comparison topology, the DRIVE-CLiQ wiring and components are checked based on the SINAMICS target topology from the DRV archive.

The following options are offered for the "comparison topology (*.ust)", which are activated when the package is executed:

List	The ... button opens the "List of topologies" dialog. Select the desired topology from the list. Comparison topologies can be inserted or linked under "List". These are also saved in the package during deployment and can be selected during package execution.
Preselection	Select a topology that is to be preselected for the package execution. Alternatively, enter the path of a file which is located next to the package "/<path>/<file>", e.g. "Comparison-topology.ust".

User-specified topology (*.utz)

Note

Prerequisite for user-specified topology

If, on the System configuration dialog page, User-specified topology has been selected under "Data source", a configured user-specified topology always applies.

If no critical inconsistency with the real topology is found by the drive in the user-defined topology, the user-defined topology is applied as the target topology by SINAMICS.

The following options are offered for the "user-specified topology (*.utz)", which are activated when the package is executed:

List	The ... button opens the "List of user-specified topologies (*.utz)" dialog. Select the desired User-specified topology (*.utz) from the list. User-specified topologies can be inserted or linked under "List". These are also saved in the package during deployment and can be selected during package execution.
Preselection	Select a user-specified topology (*.utz) that is to be preselected for the package execution. Alternatively, enter the path of a file which is located next to the package "/<path>/<file>", e.g. "User-specified-topology.utz".

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 has been selected in the "System configuration" dialog.

DO variables

The following options are offered for the "DO variables", which are activated when the package is executed:

List	The ... button opens the "Manage DO variable list" dialog. Enter the "DO variable" and its corresponding data here: ● DO type ● DO No. ● DO name ● Description

The following can be defined for a DO variable:

<table>
<tr><th>Parameter</th><th colspan="2">Description</th><th>Character</th></tr>
<tr><td>DO variable</td><td colspan="2">Unique name within the package (without specifying `Up.`)</td><td>Characters:
`A-z_0-9`</td></tr>
<tr><td rowspan="13">DO type</td><td colspan="2">Type depending on the selected category:</td><td></td></tr>
<tr><td><b>Category</b></td><td><b>Type</b></td><td rowspan="12"></td></tr>
<tr><td rowspan="3">Control Unit</td><td>Integrated (NCU)</td></tr>
<tr><td>SINAMICS</td></tr>
<tr><td>Extended (NX)</td></tr>
<tr><td rowspan="3">Line Module</td><td>ACTIVE INFEED CONTROL</td></tr>
<tr><td>SMART INFEED CONTROL</td></tr>
<tr><td>BASIC INFEED CONTROL</td></tr>
<tr><td>Motion Control</td><td>SERVO</td></tr>
<tr><td>Terminal Module</td><td>TM31</td></tr>
<tr><td>Terminal Board</td><td>TB30</td></tr>
<tr><td>Hub Module</td><td>DMC20 / DME20</td></tr>
<tr><td>CULINK</td><td>CULINK</td></tr>
<tr><td>DO No.</td><td colspan="2">DO number to be assigned to the drive object.
<b>NOTE</b>
No number may be assigned for the CU_Link DO type.</td><td>`1-54`</td></tr>
<tr><td>DO name</td><td colspan="2">DO name to be assigned to the drive object.</td><td>Characters (max. 24):
`A-z_0-9`</td></tr>
<tr><td>Description</td><td colspan="2">Additional information for the operator.</td><td>-</td></tr>
</table>

The assignment of the DO number, the DO name and the DO description is optional.

The DO variable list is used in checking manipulation tasks to ensure that only defined objects are accessed.

When a comparison or user-defined topology is applied, this list is mixed with the DO variables it contains if different DO variable lists are used in Expert and Topo.

Both in the comparison topology and in the user-specified topology, DO variables can be assigned to the SINAMICS components, which are declared in the DO variable list of Topo. By assigning a DO variable, all components are identified that belong to a drive object (e.g. a Motor Module, two Sensor Modules).

During package execution, values are assigned to the properties of the DO variables so that, for example, indirect addressing of the DOs for reading and writing parameters can be performed.

If neither a comparison topology nor a target topology is defined, the DO variables from the DO variable list configured in Expert can be assigned manually.

The DO variable list can be exported, imported, or linked. We recommend using the same list in Expert and Topo. The easiest way to accomplish this is to externally save this list and link it in Expert and in Topo. The list can then be viewed and edited by both Expert and Topo.

Note

For DO manipulations using a comparison topology, user-specified topology, or DO list

When changing the DO number, the assigned PS file is renamed and parameters p101 and p978 of the higher-level CU are adapted.
The DO name is entered in parameter p199 of the relevant DO, and the DO variable name in parameter p9489.
BiCos are not automatically adapted. This should be noted particularly for the Line Module.

Note

Conflicts between DO variables as regard type, number, or name result in errors.

Note

The TM31 Terminal Module is not supported by CNC SW 4.4 SP1 with Create MyConfig.

Dialog scripts

Under "Dialog scripts", entry "Scripts", you open the "ScriptEditor dialog - SINAMICS topologyjack (<time of execution>)" dialog page via the ... button.
You can edit scripts here that are to be executed at selectable times during the package execution.

In this dialog, the specified dialog scripts of selected dialogs can be assigned to the desired events (=time) without having to call each individual dialog.

Assignment dialog - Event

The dialogs in which dialog scripts can be assigned to selected times (events), are listed in the Assignment dialog - Event (Page 101) table.

Meaning of the events:

Possible events and their meaning can be found in the Event - Meaning (Page 102) table.

Inserting a script

Information on inserting a script and on the following options can be found in the Scripting (Page 197) section under:

- Sections (Page 204)
- Area data (Page 206)
- Check structures (Page 206)
- Operators (Page 209)
- Functions (Page 212)
- Procedures (Page 231)

Package execution

Using the "Topology file" 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 (with comparison topology, e.g. XML generated by E-CAD system) can be selected with the "Browse..." button.

Comparison topology

In the interactive assignment of the DO variables 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 association of the sensors to the axes/drives is also defined.

In the comparison topology, additional information can be hidden or displayed for diagnostics in the list and tree view using the "Display additional details" checkbox.

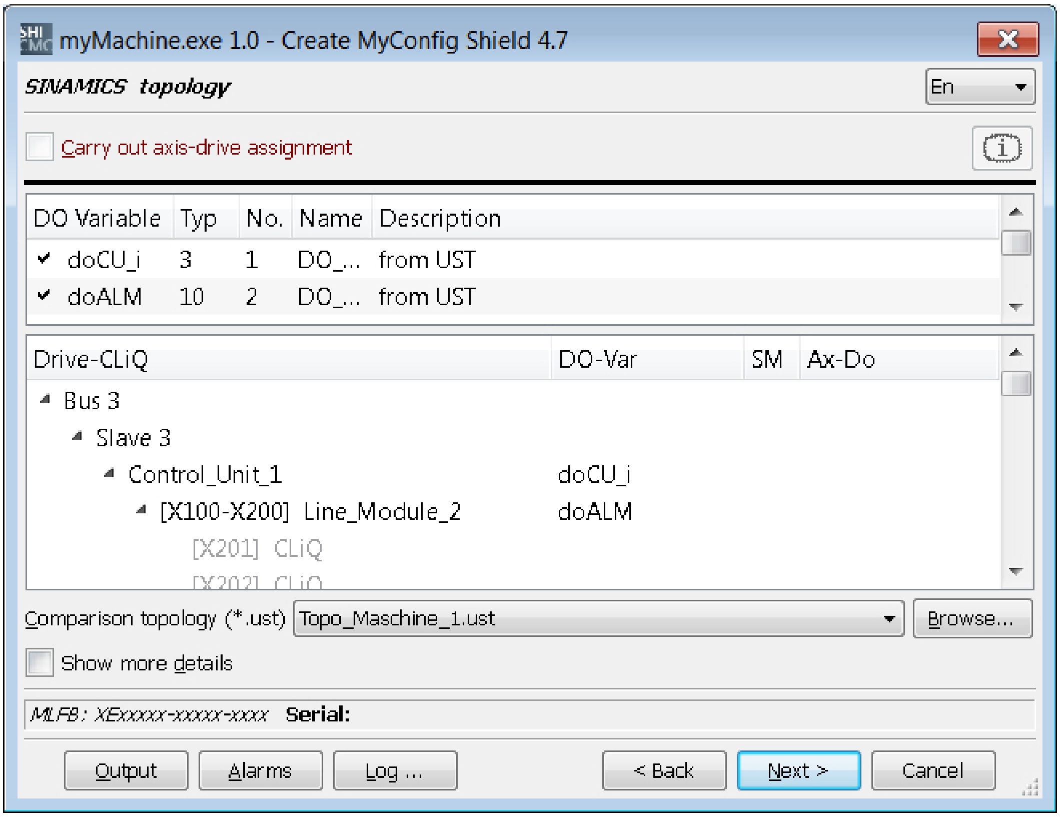
User-specified topology

The three different activation states of the SINAMICS modules "0-inactive," "1-active," and "2-not present" can be adapted interactively. With a click on the relevant cell, the three states are connected.
Keyboard operation is performed with UP/Down in the line and then "-DO" of the "K component".

Dialog view during the package execution (for commissioning engineers / operators)

The "SINAMICS topology" dialog page is displayed here with an example of a **Comparison topology**:

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



☐ Carry out axis-drive assignment

	DO Variable	Typ	No.	Name	Description
✓	doCU_i	3	1	DO_...	from UST
✓	doALM	10	2	DO_...	from UST

Drive-CLiQ	DO-Var	SM	Ax-Do
◢ Bus 3			
◢ Slave 3			
◢ Control_Unit_1	doCU_i		
◢ [X100-X200] Line_Module_2	doALM		
[X201] CLiQ			
[X202] CLiQ			

Comparison topology (*.ust) Topo_Maschine_1.ust

☐ Show more details

MLFB: XExxxxx-xxxxx-xxxx **Serial:**

Figure 4-26 SINAMICS topology dialog - commissioning engineer / operator view

Buttons in dialogs

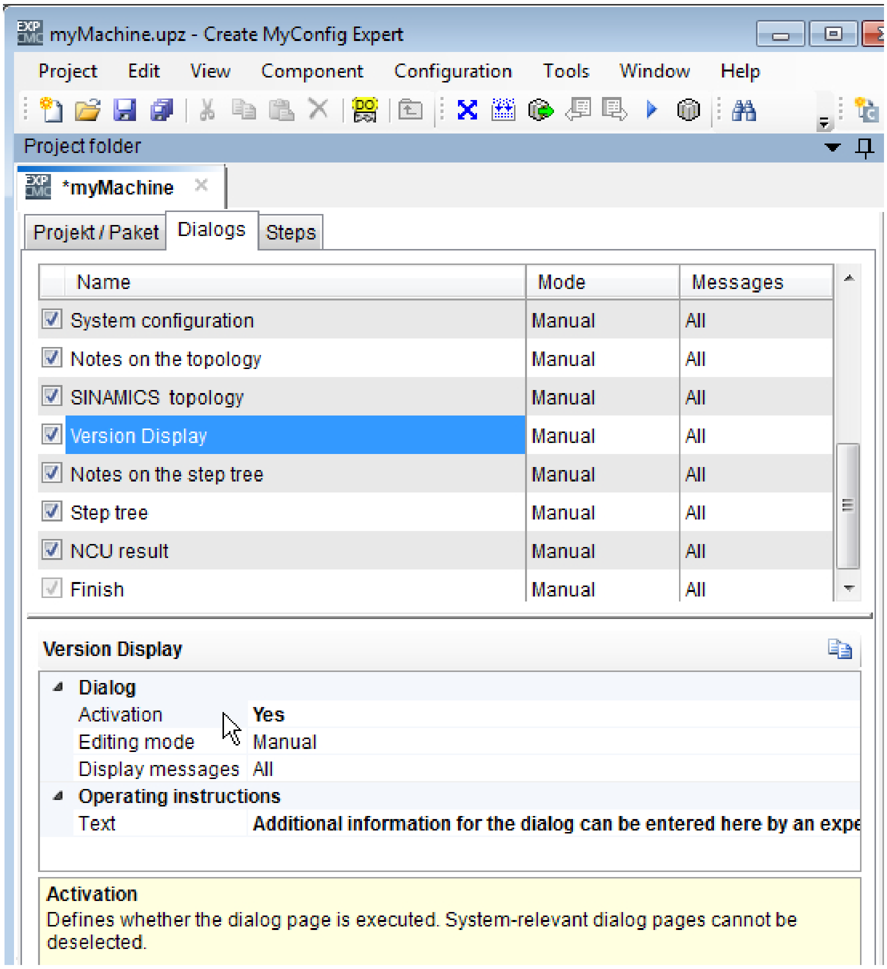
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.13 Dialog: Version display

Configuration

You can configure the following property groups for the "Version Display" dialog page:

- Dialog
- Operating instructions



Name	Mode	Messages
☑ System configuration	Manual	All
☑ Notes on the topology	Manual	All
☑ SINAMICS topology	Manual	All
☑ Version Display	Manual	All
☑ Notes on the step tree	Manual	All
☑ Step tree	Manual	All
☑ NCU result	Manual	All
☑ Finish	Manual	All

Version Display

Dialog	
Activation	**Yes**
Editing mode	Manual
Display messages	All
Operating instructions	
Text	**Additional information for the dialog can be entered here by an expe**

Activation
Defines whether the dialog page is executed. System-relevant dialog pages cannot be deselected.

Figure 4-27 Version display - project engineer view

Display all system variables via a dialog button

You open the "Version display - System variables" dialog via the button (display all system variables via a dialog). All system variables to be used under "Version display" are listed here.

Information on the system variables·can be found in the System variables for the Version display dialog (Page 279) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

Dialog view during the package execution (for commissioning engineers / operators)

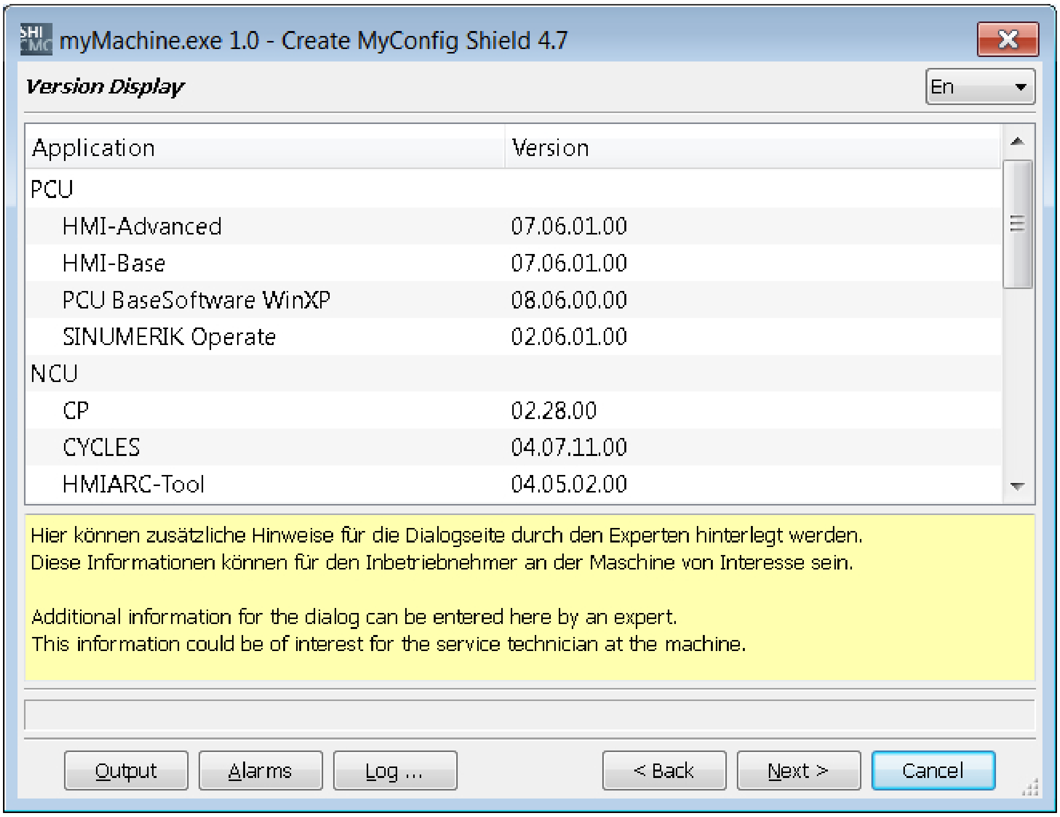
The applications identified during execution and their determined versions are displayed on the "Version display" dialog.

For the NCU, the information is obtained from the file: "/siemens/versions.xml" and for the PCU from the registry: "HKLM\Software\SIEMENS\AUTSW".

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).

Note

For the version query (see: Steps (Page 166)) for the conditional step execution or in the scripting, the names displayed here should be used as keywords. The names can also be shortened as long as they remain unique.



Application	Version
PCU	
HMI-Advanced	07.06.01.00
HMI-Base	07.06.01.00
PCU BaseSoftware WinXP	08.06.00.00
SINUMERIK Operate	02.06.01.00
NCU	
CP	02.28.00
CYCLES	04.07.11.00
HMIARC-Tool	04.05.02.00

Hier können zusätzliche Hinweise für die Dialogseite durch den Experten hinterlegt werden.
Diese Informationen können für den Inbetriebnehmer an der Maschine von Interesse sein.

Additional information for the dialog can be entered here by an expert.
This information could be of interest for the service technician at the machine.

Figure 4-28 Version Display dialog - commissioning engineer / operator view

Buttons in dialogs

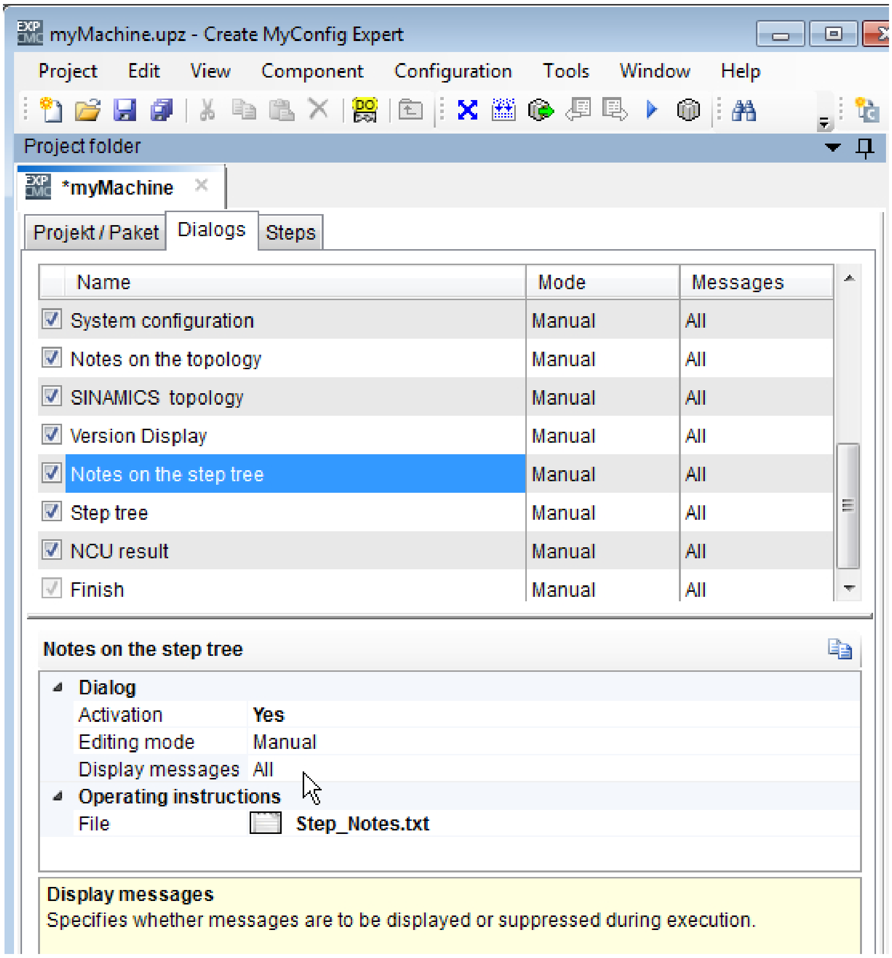
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.14 Dialog: Notes on the step tree

Configuration

You can configure the following property groups for the "Notes on the step tree" dialog page:

- Dialog
- Operating instructions



	Name	Mode	Messages
☑	System configuration	Manual	All
☑	Notes on the topology	Manual	All
☑	SINAMICS topology	Manual	All
☑	Version Display	Manual	All
☑	Notes on the step tree	Manual	All
☑	Step tree	Manual	All
☑	NCU result	Manual	All
☑	Finish	Manual	All

Notes on the step tree

Dialog

Activation	**Yes**
Editing mode	Manual
Display messages	All

Operating instructions

File	**Step_Notes.txt**

Display messages
Specifies whether messages are to be displayed or suppressed during execution.

Figure 4-29 Notes on the step tree - project engineer view

Display all system variables via a dialog button

You open the "Notes on the step tree - system variables" dialog via the button 📋 (display all system variables via a dialog). All system variables to be used under "Notes on the step tree" are listed here.

Information on the "Notes on the step tree - System variables" can be found in section System variables for Up.$Dialog.StepSelectionNotes (Page 280).

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

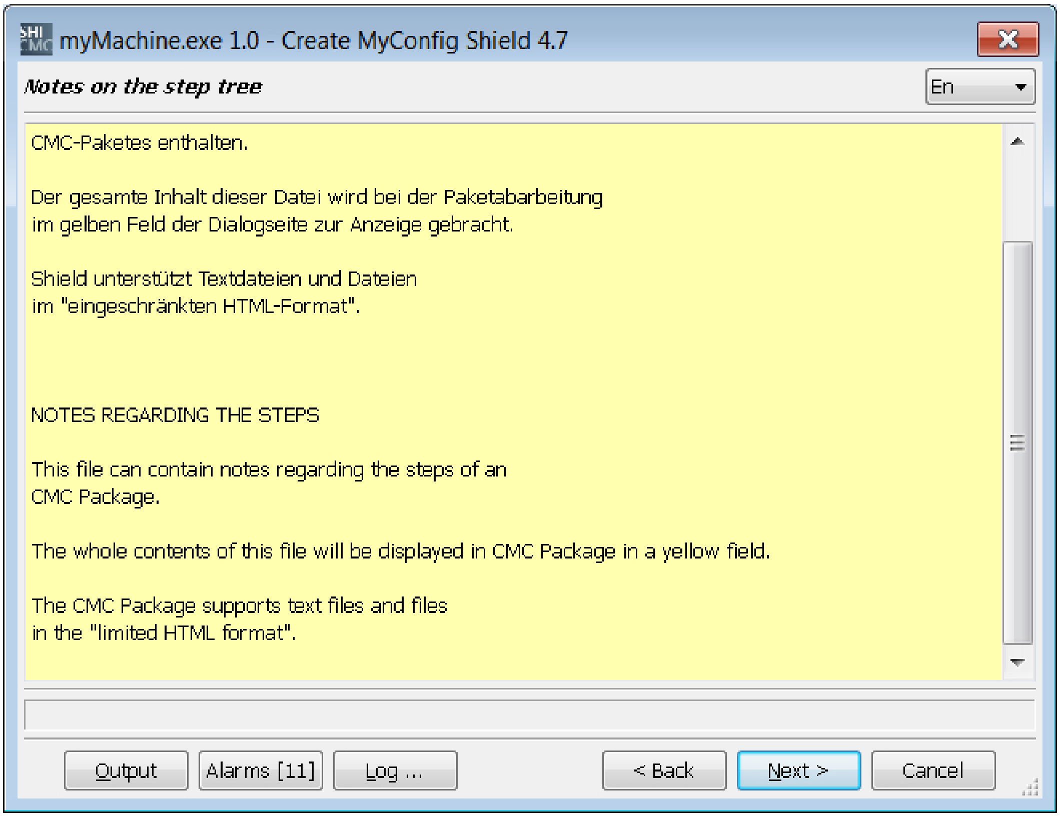
Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

Dialog view during the package execution (for commissioning engineers / operators)

A file in text format or restricted HTML format with operating instructions for the following dialog can be displayed on the "Notes on the step tree" dialog.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



CMC-Paketes enthalten.

Der gesamte Inhalt dieser Datei wird bei der Paketabarbeitung
im gelben Feld der Dialogseite zur Anzeige gebracht.

Shield unterstützt Textdateien und Dateien
im "eingeschränkten HTML-Format".

NOTES REGARDING THE STEPS

This file can contain notes regarding the steps of an
CMC Package.

The whole contents of this file will be displayed in CMC Package in a yellow field.

The CMC Package supports text files and files
in the "limited HTML format".

Figure 4-30 Notes on the step tree dialog - commissioning engineer / operator view

Buttons in dialogs

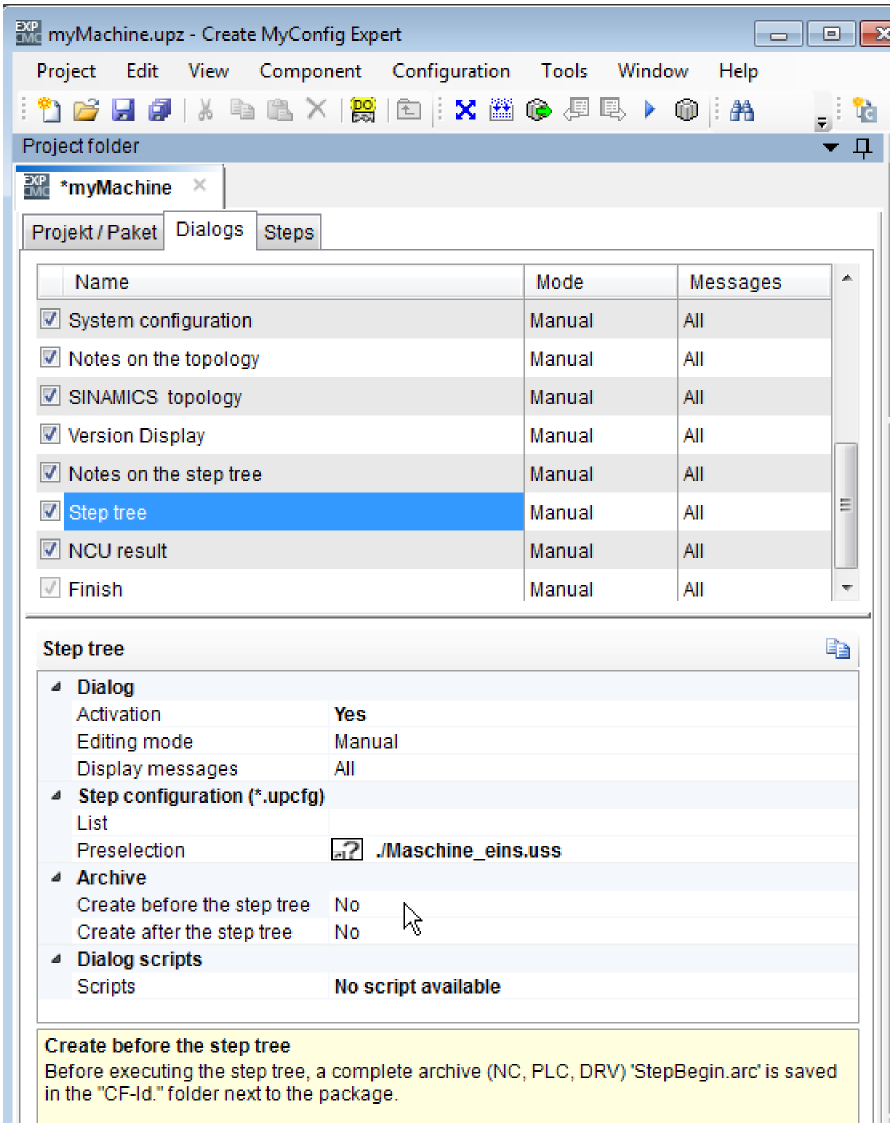
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.15 Dialog: Step tree

Configuration

You can configure the following property groups for the "Step tree" dialog page:

- Dialog
- Step configuration (*.uss)
- Archive
- Dialog scripts



	Name	Mode	Messages
☑	System configuration	Manual	All
☑	Notes on the topology	Manual	All
☑	SINAMICS topology	Manual	All
☑	Version Display	Manual	All
☑	Notes on the step tree	Manual	All
☑	Step tree	Manual	All
☑	NCU result	Manual	All
☑	Finish	Manual	All

Dialog	
Activation	**Yes**
Editing mode	Manual
Display messages	All
Step configuration (*.upcfg)	
List	
Preselection	**./Maschine_eins.uss**
Archive	
Create before the step tree	No
Create after the step tree	No
Dialog scripts	
Scripts	**No script available**

Create before the step tree
Before executing the step tree, a complete archive (NC, PLC, DRV) 'StepBegin.arc' is saved in the "CF-Id." folder next to the package.

Figure 4-31 Step tree - project engineer view

Display all system variables via a dialog button

You open the "Step tree - System variables" dialog via the button 📋 (display all system variables via a dialog). All system variables to be used under "Step tree" are listed here.

Information on the system variables·can be found in the System variables for the Step tree dialog (Page 280) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog

More detailed information about the property group "Dialog" can be found under the dialog page: Notes about acknowledgement (Page 104).

Step configuration (*.upcfg)

The following settings are available for the "Step configuration (*.upcfg)" which are activated when the package is executed:

List	The ... button opens the "List of step configurations (*.upcfg)" dialog. Under "List", step configurations (*.upcfg and/or *.uss) can be inserted or linked. These are also saved in the package during deployment and can be selected during package execution. Such a list can be exported in the project folder in the Steps tab (Page 166).
Preselection	Select a step configuration that is to be preselected for the package execution.

Further information can be found in the Interface - Step-XML (Page 269) section.

Archive

The settings for the creation of archives before and after execution of the step tree are specified by the project engineer and not displayed during the package execution.

The archives can be used for the diagnostics. They are not required for the production or the upgrade of a machine.

The following settings are available for "Archive":

Create before step tree	Yes	Here, you specify whether an archive with the NC, PLC and DRV areas is to be created **before** the execution of the step tree. The backup is performed for the NC area with the setting that all data is to be saved. With the setting "Yes", the archive is saved with the name '**StepBegin_**<date>.arc'. The archive is stored in the folder based on the name of the CF card ID which is located next to the package after the package execution.
	No	

Create after step tree	Yes	Here, you specify whether an archive with the NC, PLC and DRV areas is to be created **after** the execution of the step tree. The backup is performed for the NC area with the setting that all data is to be saved. With the setting "Yes", the archive is saved with the name '**StepEnd_**<date>.arc'. The archive is stored in the folder based on the name of the CF card ID which is located next to the package after the package execution.
	No	

Dialog scripts

Under "Dialog scripts", entry "Scripts", you open the "ScriptEditor dialog - Step tree (<time of execution>)" dialog page via the ... button.
You can edit scripts here that are to be executed at selectable times during the package execution.

In this dialog, the specified dialog scripts of selected dialogs can be assigned to the desired events (=time) without having to call each individual dialog.

Assignment dialog - Event

The dialogs in which dialog scripts can be assigned to selected times (events), are listed in the Assignment dialog - Event (Page 101) table.

Meaning of the events:

Possible events and their meaning can be found in the Event - Meaning (Page 102) table.

Inserting a script

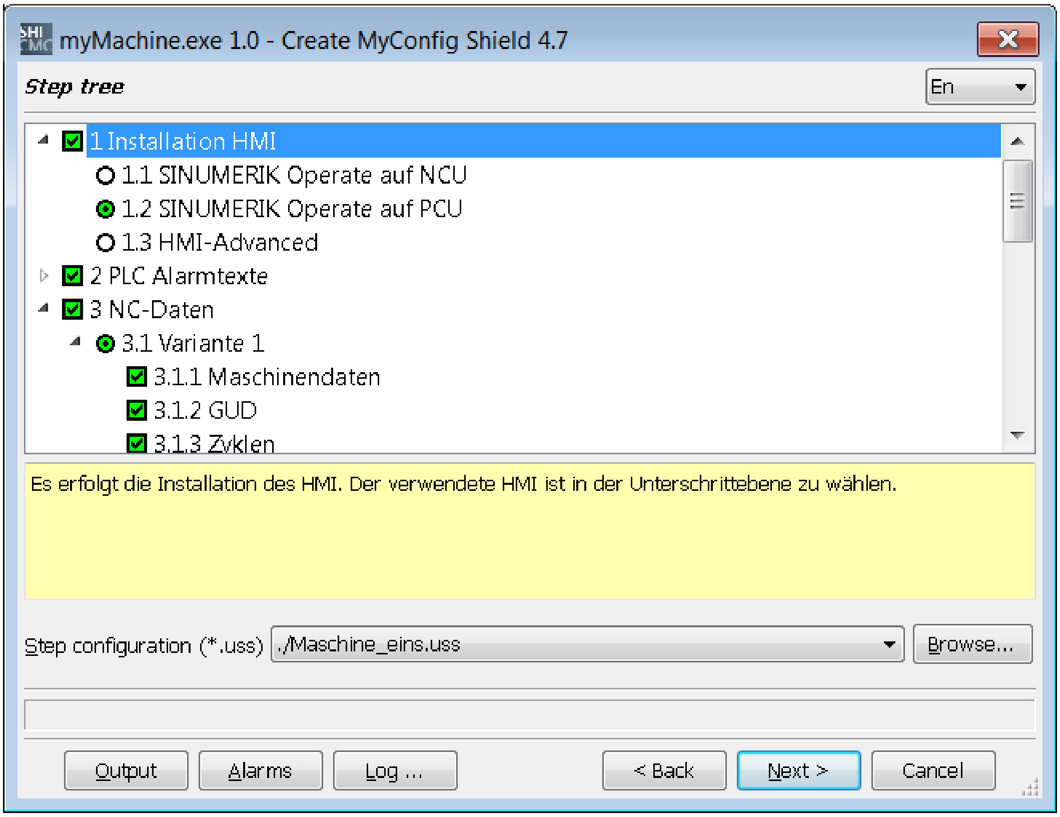
Information on inserting a script and on the following options can be found in the Scripting (Page 197) section under:

- Sections (Page 204)
- Area data (Page 206)
- Check structures (Page 206)
- Operators (Page 209)
- Functions (Page 212)
- Procedures (Page 231)

Dialog view during the package execution (for commissioning engineers / operators)

On the dialog page, the operator can activate or deactivate interactive steps. The information for the selected step is displayed at the bottom in the yellow field and makes it easier for the operator to understand.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



Es erfolgt die Installation des HMI. Der verwendete HMI ist in der Unterschrittebene zu wählen.

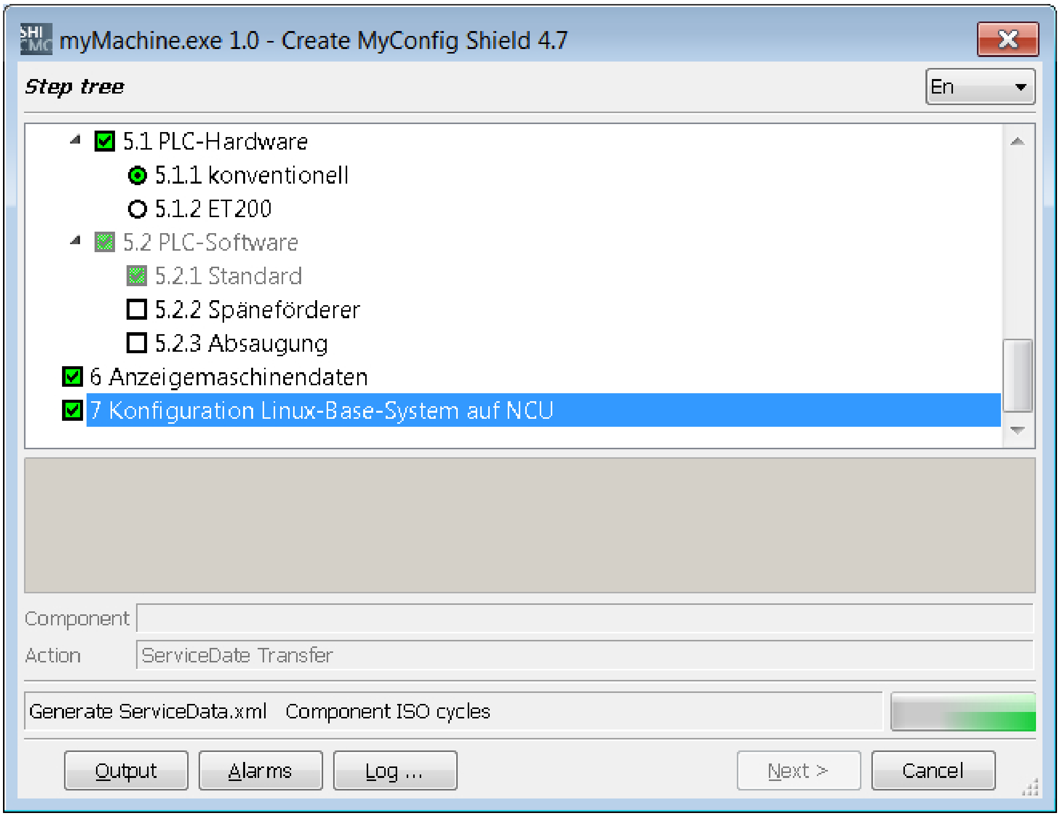
Step configuration (*.uss) ./Maschine_eins.uss

Figure 4-32 1. Step tree dialog - commissioning engineer / operator view

Using the "Step configuration (*.upcfg)" drop-down list, a file that was assigned to the package or a file located next to the package (prefix: "./") can be used. The "Browse" button allows a selection to be made from any location.

In addition to files of the type *.upcfg, files of the type *.uss can also be used.

The preparation of the execution is confirmed with the "Next" button and it is not possible to return.



☑ 5.1 PLC-Hardware
● 5.1.1 konventionell
○ 5.1.2 ET200
☑ 5.2 PLC-Software
☑ 5.2.1 Standard
□ 5.2.2 Späneförderer
□ 5.2.3 Absaugung
☑ 6 Anzeigemaschinendaten
☑ 7 Konfiguration Linux-Base-System auf NCU

Component:
Action: ServiceDate Transfer

Generate ServiceData.xml   Component ISO cycles

Figure 4-33 2. Step tree dialog - commissioning engineer / operator view

Buttons in dialogs

Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

See also

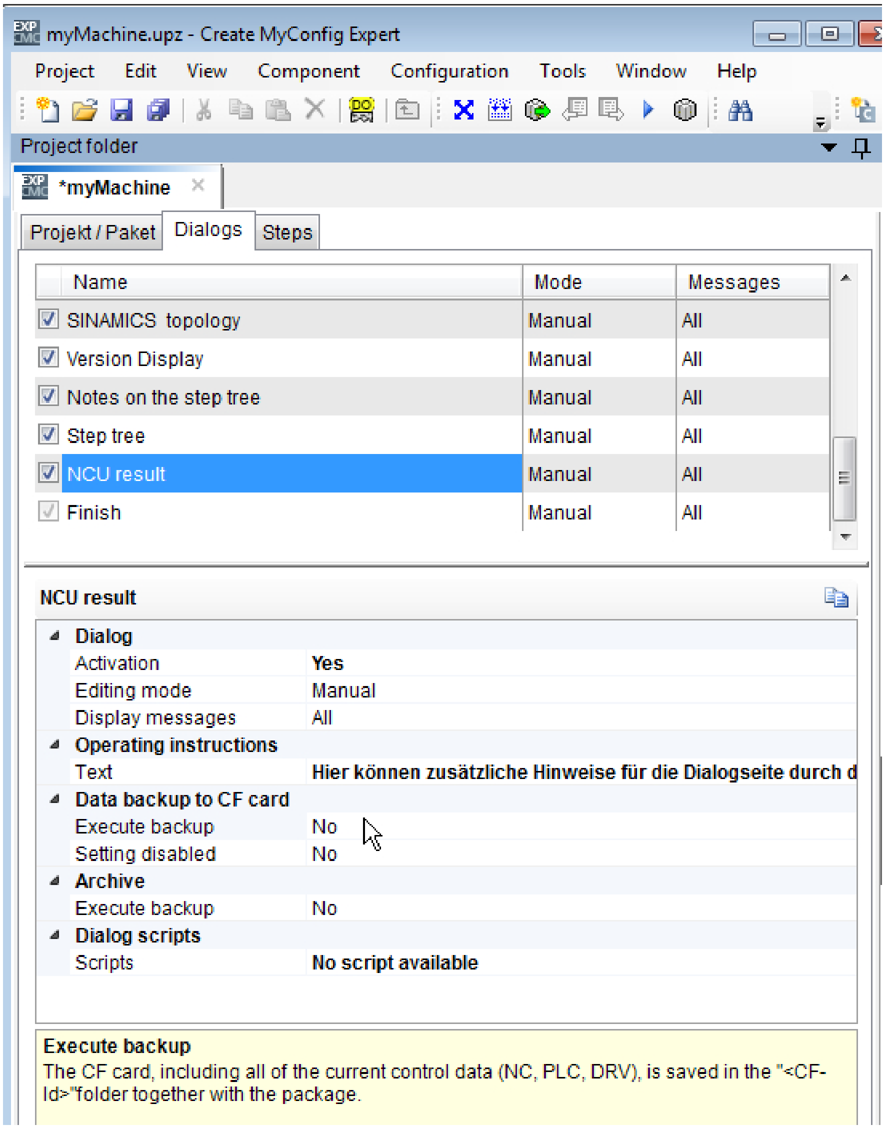
Steps (Page 166)

4.3.2.16 Dialog page: NCU result

Configuration

You can configure the following property groups for the "NCU result" dialog page:

- Dialog
- Operating instructions
- Data backup CF card
- Archive
- Dialog scripts



Name	Mode	Messages
☑ SINAMICS topology	Manual	All
☑ Version Display	Manual	All
☑ Notes on the step tree	Manual	All
☑ Step tree	Manual	All
☑ NCU result	Manual	All
☑ Finish	Manual	All

NCU result

Dialog	
Activation	**Yes**
Editing mode	Manual
Display messages	All
Operating instructions	
Text	**Hier können zusätzliche Hinweise für die Dialogseite durch d**
Data backup to CF card	
Execute backup	No
Setting disabled	No
Archive	
Execute backup	No
Dialog scripts	
Scripts	**No script available**

Execute backup
The CF card, including all of the current control data (NC, PLC, DRV), is saved in the "<CF-Id>"folder together with the package.

Figure 4-34 NCU result - Project engineer view

Display all system variables via a dialog button

You open the "NCU result - System variables" dialog via the button 📋 (display all system variables via a dialog). All system variables to be used under "NCU access" are listed here.

Information on the system variables·can be found in the System variables for the NCU result dialog (Page 281) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

Data backup CF card

The "NCU result" dialog page is used to back up all of the data of the NCU which runs under the Linux operating system.

The following settings are available for the "Data backup CF card" and are activated when the package is executed:

Perform backup	Yes	Here, you specify whether a data backup of the CF card including the archive data for the NC, PLC and DRV areas is to be created after the package execution. With the setting "Yes", the data backup is saved with the name '<CF card ID>_result.tgz'. The file is stored in the folder based on the CF card ID which is located next to the package after the package execution.
	No	
Setting disabled	Yes	Here, you specify whether the settings made during the configuration are to be blocked so that they can only be viewed by the operator, but not changed.
	No	

Archive

The settings for the archive are defined by the project engineer and are not displayed during package execution.

The archive can be used for the diagnostics. It is not required for the production or the upgrade of a machine.

The following settings are available for the "Archive":

Perform backup	Yes	Here, you specify whether an archive with the NC, PLC and DRV areas is to be created **after** the execution of the package. The backup is performed for the NC area with the setting that all data is to be saved. With the setting "Yes", the archive is saved with the name 'Result_<date>.arc'. The archive is stored in the folder based on the name of the CF card ID which is located next to the package after the package execution.
	No	

Dialog scripts

Under "Dialog scripts", entry "Scripts", you open the "ScriptEditor dialog - NCU result (<time of execution>)" dialog page via the ... button.
You can edit scripts here that are to be executed at selectable times during the package execution.

In this dialog, the specified dialog scripts of selected dialogs can be assigned to the desired events (=time) without having to call each individual dialog.

Assignment dialog - Event

The dialogs in which dialog scripts can be assigned to selected times (events), are listed in the Assignment dialog - Event (Page 101) table.

Meaning of the events:

Possible events and their meaning can be found in the Event - Meaning (Page 102) table.

Inserting a script

Information on inserting a script and on the following options can be found in the Scripting (Page 197) section under:

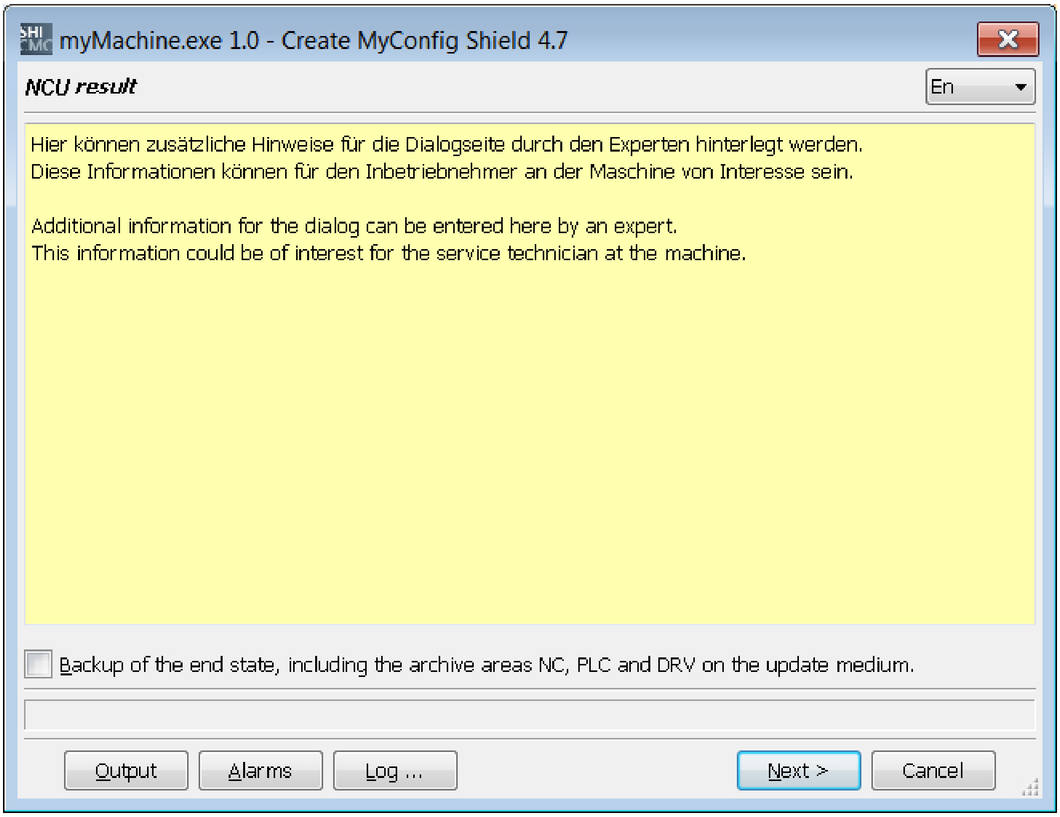
- Sections (Page 204)
- Area data (Page 206)
- Check structures (Page 206)
- Operators (Page 209)
- Functions (Page 212)
- Procedures (Page 231)

Dialog view during the package execution (for commissioning engineers / operators)

A data backup of the CF card including all current control data (NC, PLC, DRV) can be selected on this dialog page.

The data backup is stored with the name '<C cardID>_result.tgz' in the folder based on the CF card ID which is located next to the package after the package execution.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).



Hier können zusätzliche Hinweise für die Dialogseite durch den Experten hinterlegt werden.
Diese Informationen können für den Inbetriebnehmer an der Maschine von Interesse sein.

Additional information for the dialog can be entered here by an expert.
This information could be of interest for the service technician at the machine.

□ Backup of the end state, including the archive areas NC, PLC and DRV on the update medium.

Figure 4-35 NCU result dialog - commissioning engineer / operator view

Buttons in dialogs

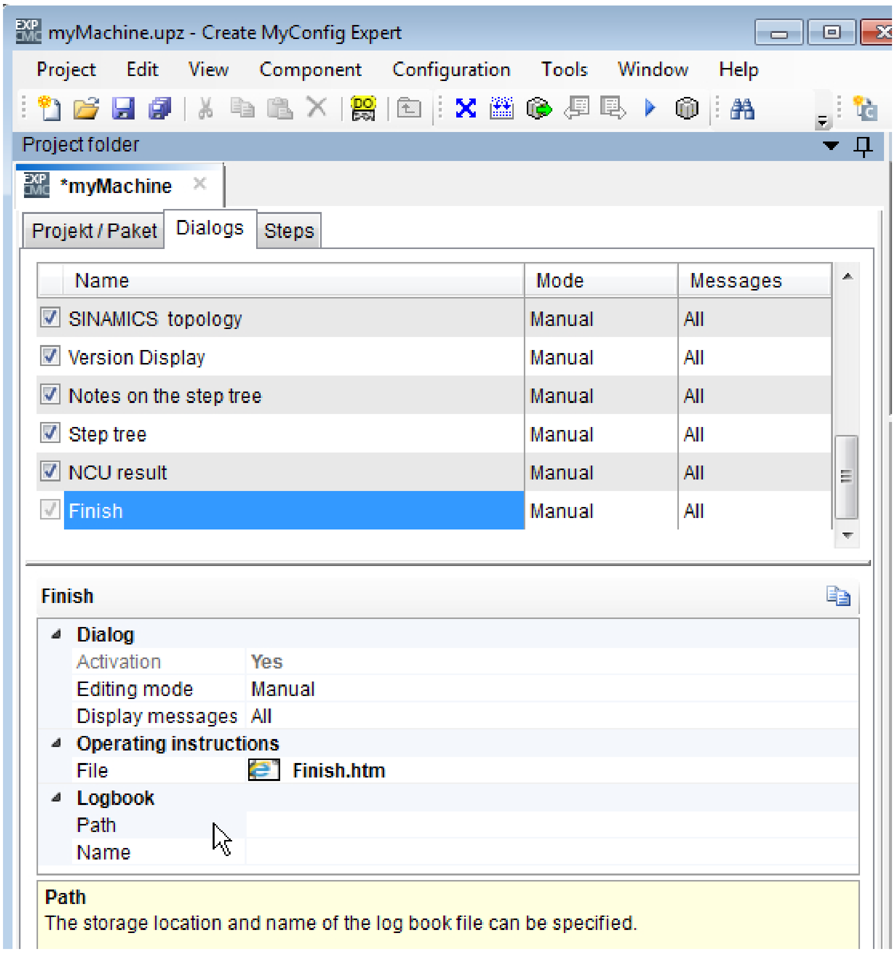
Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.2.17 Dialog: Finish

Configuration

You can configure the following property groups for the "Finish" dialog page:

- Dialog
- Operating instructions
- Logbook



	Name	Mode	Messages
☑	SINAMICS topology	Manual	All
☑	Version Display	Manual	All
☑	Notes on the step tree	Manual	All
☑	Step tree	Manual	All
☑	NCU result	Manual	All
☑	Finish	Manual	All

Finish

Dialog	
Activation	Yes
Editing mode	Manual
Display messages	All
Operating instructions	
File	**Finish.htm**
Logbook	
Path	
Name	

Path
The storage location and name of the log book file can be specified.

Figure 4-36 Dialog page: Finish - project engineer view

Display all system variables via a dialog button

You open the "Finish - System variables" dialog via the button (display all system variables via a dialog). All system variables to be used under "Finish" are listed here.

Information on the system variables·can be found in the System variables for the Finish dialog (Page 281) section.

Further general information on the system variables can be found in the Reference list system variables (Page 271) section.

Dialog / Operating instructions

More detailed information about the property groups "Dialog" and "Operating instructions" can be found under the dialog page: Notes about acknowledgement (Page 104).

Logbook

The following settings are available for the "Logbook" and are activated when the package is executed:

Path	Here, you define the storage location of the logbook.
Name	Here, you specify the name of the logbook.

Note

Storage location logbook

If a storage path has not been configured, the logbook is stored next to the package.

For the package execution with the **NCU** area, the logbook is also stored on the control under /card/install.

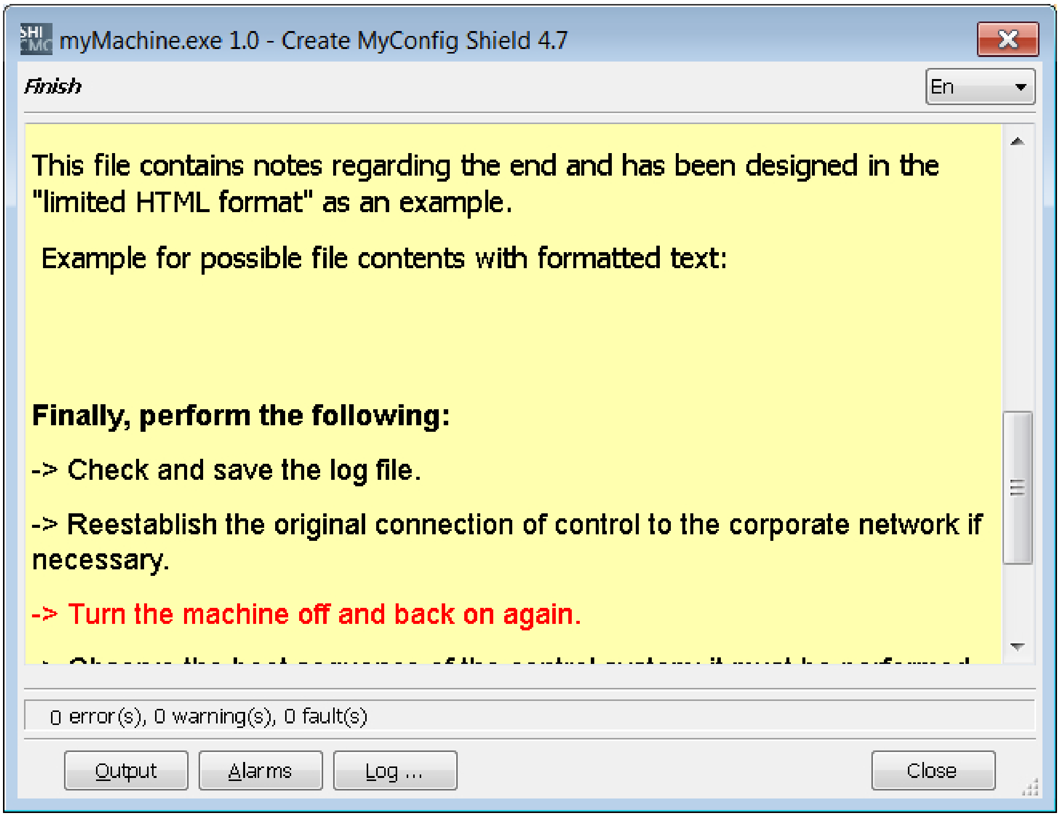
For a package which runs on the **PCU** or Windows PC, it is stored under:
d:\UP of the PCU if a PCU is involved.

Dialog view during the package execution (for commissioning engineers / operators)

The "Finish" dialog cannot be deactivated. A file in text format or restricted HTML format can be displayed on this dialog for postprocessing.

The operator can also be prompted to save the logbook when closing the dialog. A logbook that can only be opened after a password has been entered is stored as an *.exe file.

You will find additional information on package execution in the section under Create MyConfig packages (Page 246).


myMachine.exe 1.0 - Create MyConfig Shield 4.7
Finish
En
0 error(s), 0 warning(s), 0 fault(s)
Output
Alarms
Log ...
Close

This file contains notes regarding the end and has been designed in the "limited HTML format" as an example.

Example for possible file contents with formatted text:

Finally, perform the following:

-> Check and save the log file.

-> Reestablish the original connection of control to the corporate network if necessary.

-> Turn the machine off and back on again.

Figure 4-37 Finish dialog - commissioning engineer / operator view

Buttons in dialogs

Information on the general buttons in the dialogs can be found under Meaning of buttons (Page 103).

4.3.3 Steps

On the "Steps" tab, you can create a hierarchically structured step tree.

Structuring of the steps

Because of the structuring of the steps on several levels into main steps and substeps and the possibility of activating/deactivating the individual steps, the different variants of the machines of the series can be modeled by the step tree.

Properties are defined for each step during configuration. The control elements for activation of the steps can be configured both for the **m** out of **n** selection and for the **1** out of **n** selection. The step activations that you have specified can also be changed by the operator on the machine during the package execution.

Package execution

When executing the package, the step tree is executed from top to bottom along the **step sequence marked in green**. A step is always executed if its checkbox is green in the display. When a step is executed, any existing step script is first checked. If the step script defines that the step is to be processed further, the components linked to the step will be executed in the configured sequence. Execution then proceeds to the next step.

When generating the step tree using the context menu, steps should be arranged in the sequence that they should be executed when executing the package.
A step name can be freely selected. The numbering has no functional significance and is only used to improve the overall orientation.

The step tree is simultaneously used for navigation when processing/editing components.

Steps can be copied, cut and pasted by means of the context menu. This covers all the properties of a step and its subordinate steps (conditions, component links, notes, etc.).

Note

The external components allocated to the step are linked and are not copied. The link remains.

If steps with internal components are copied between projects, a selection can be made between "Paste" and "Insert as link".

If "Paste" is selected, a copy is inserted as internal component in the target project.

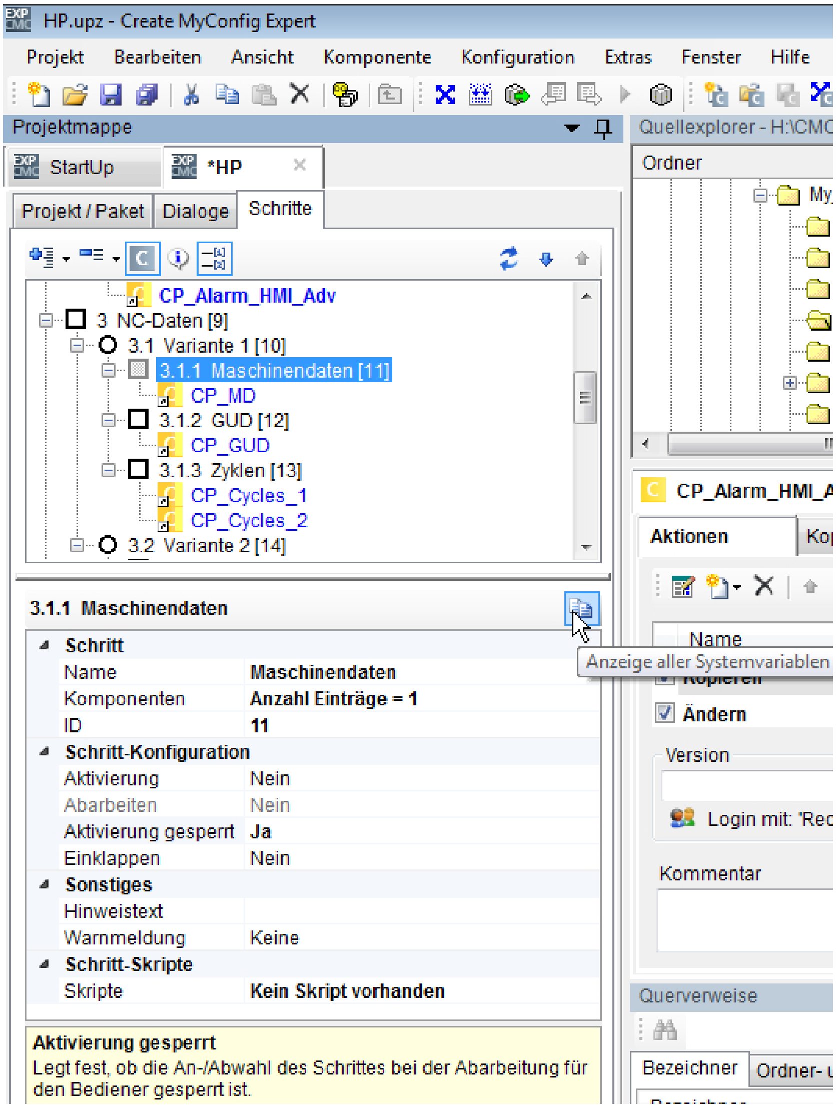
If "Insert as link" is selected, the component remains in the original project and is linked as an external component.

4.3.3.1 Step tree structure

Step tree structure

The step tree has a hierarchical structure encompassing:

- Steps
- Substeps
- Components



3.1.1 Maschinendaten

Schritt	
Name	**Maschinendaten**
Komponenten	**Anzahl Einträge = 1**
ID	**11**
Schritt-Konfiguration	
Aktivierung	Nein
Abarbeiten	Nein
Aktivierung gesperrt	**Ja**
Einklappen	Nein
Sonstiges	
Hinweistext	
Warnmeldung	Keine
Schritt-Skripte	
Skripte	**Kein Skript vorhanden**

Aktivierung gesperrt
Legt fest, ob die An-/Abwahl des Schrittes bei der Abarbeitung für den Bediener gesperrt ist.

Figure 4-38 Example Structure of a step tree

Buttons for the step tree

The display options of the Steps tab are described in the following.

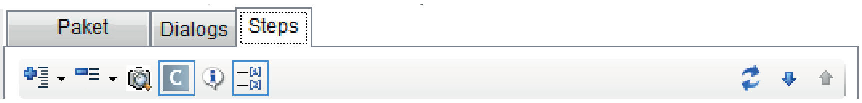


Figure 4-39 Toolbar of the Steps tab

Using the buttons, you can modify the representation of the step tree:

▾	**Expand selection** Here, you can either display a selected step or the entire step tree with all substeps and components.
▾	**Reduce selection** Here, you can reduce an individually selected step or the entire step tree to the main step. Substeps and components are thus no longer visible.
	Preview for step tree during the package execution The project engineer can check the operator view here. Collapsed steps are not displayed.
C	**Show components** The components for the displayed steps are displayed or hidden via this button.
ⓘ	**Display tooltip for steps** Select this option to display tooltips for steps or substeps. Any existing warning messages are also displayed in the tooltip.
—[1] —[2]	**Display step ID** The step IDs are displayed or hidden via this button.
⟳	**Refresh step tree completely (F5)** Via this button, **all** of the changes that were made in the entire step tree to steps, substeps and components are refreshed in the display. **Meaning of the colored display of a link step**: ● **Blue** The update was successful. ● **Red** The update could not be performed because the subproject cannot be reached. ● **Purple** During the update, a difference was found in the main project; the step configuration must be checked. (Purple display only possible for link step that is managed in the main project.) **Note** The changes are **not** saved by the update.
↓ ↑	**Down / Up** Using the arrow keys, you can change the sequence of selected steps, substeps or components which are located on a level.

"Display all system variables via a dialog" button

You open the "System variables" dialog via the button (display all system variables via a dialog).

The "System variables" dialog shows the list of system variables Up.$Step[<id>] for the currently selected step.

Information on the system variables can be found under the System variables reference list in the Steps tab (Page 282) section.

4.3.3.2 Step properties

Configuration

You can configure the following property groups for each step or substep:

- Step
- Step configuration
- Miscellaneous
- Step scripts

Step

The following settings are available for the property group "Step":

Name	A name can be assigned to each step. This name is not relevant for the functioning, but can help the expert in Expert and the operator on the machine with regard to content and meaning of the step.
Components	Use the button ... to open the "Component list" dialog. The available components are listed in the "Component list" dialog. You can change their order using the arrow keys ↓ ↑ on the top right. You can link new or existing component(s) via the buttons top left. The "Component Explorer" dialog opens for this purpose. Here you can create new components or link existing components. Components can also be deleted from the Component list. You can "Delete the link from the step" ✕; this results in the component being retained and it can be linked again via the Component explorer or the component is irrevocably deleted ✕. The components are executed according to the order of their listing from top to bottom when executing the package, provided that the step activation is "green" and the step execution is not skipped due to step scripts. No components must be linked to steps. Steps can also be created for organizational reasons only, e.g. for querying conditions, etc.
ID namespace	The ID namespace is only displayed if a step from a subproject was linked into a main project. The newly linked step is assigned a unique namespace in the main project. The designation of the namespace can be changed. Further information on the namespace (Page 177).
ID	When it is created, a step is assigned an ID which is unique in the project. This can be subsequently adapted. Either numeric values or alphanumeric values starting with a letter or '_' are permitted. While executing the step, the activation state of the step can be queried (from other steps or in the scripting) using its ID. Further information on the step activation and ID can be found in the System variables reference list - Steps tab (Page 282) section.

Step configuration

The following settings are available for the property group "Step configuration":

Management in the	Main project Subproject	Specifies where the step configuration is to be managed.
Activation	Yes No	The "Activation" defines whether a step will be called for execution. With "No" this step and all of its substeps are not executed. All steps to be executed according to this logic are identified by a green activation symbol. **Note** If a step cannot be executed due to the step scripts, the linked components of this step and the substeps will also not be executed. Because this condition can only be evaluated during the package execution, the activation symbols of the associated substeps can still be green.
Execute	Yes No	If "Execute > Yes" is displayed, the step is executed during the package execution, including the stored step scripts. An activated step, which is displayed in green in the step tree, contains the property "Execute > Yes". If "Execute > No" is displayed, a step is either not activated or the green line was interrupted due to the lack of activation in a higher-level step. This step is not considered during package execution.
Activation disabled	Yes No	If the "Activation is disabled", then when executing the package, the operator can no longer activate or deactivate the step. The context menu allows the selected step to be disabled, including all of the substeps. When executing the package, steps that only contain disabled substeps are collapsed.
Collapse	Yes No	For the time of package execution, it can be defined for each step whether the step will be collapsed. Substeps can be hidden in this way. The overview for display of the step tree can therefore be configured individually. In the default setting "No", substeps during package execution are displayed. The setting "Yes" hides the substeps. In the configuration view, collapsed steps are displayed in italics.

Miscellaneous

The following settings are available for the property group "Miscellaneous":

Information text	An information text can be entered for each step; this text is then displayed on the "Step tree" dialog if the particular step is selected or being processed.
Warning message	Optionally, a warning message can be configured which informs the operator that the step is activated or deactivated when clicking the "Finished" button on the "Step tree" dialog page.

	None	No warning message is output.
	If step activated	A warning message is output if the step is activated.
	If step not activated	A warning message is output if the step is not activated. The operator then has the option of confirming the setting using "OK" – or after "Cancel" – can correct the step activation before execution is continued.

Step scripts

The following settings are available for the property group "Step scripts":

Scripts	You can call the "ScriptEditor Script - <Number and name of the step> (OnProc)" dialog page here via the <...> button. A script can be inserted which is executed before the execution of the step. For example, a step condition can be programmed in this step script.

4.3.3.3 Edit step tree

Component editor

Double-clicking a component link opens the component in the component editor. Open components are shown in **bold** in the step tree.

A single click on a component that is already open, brings the associated component editor into the foreground.

With the C symbol, internal components and with the C symbol, external components are displayed.

The content of the function of internal and external components is completely identical. They differ only be their storage location. Internal components are stored in the project folder and external components are stored in any accessible folder. External components can therefore be processed independently of the project.

Context menu

Using the context menu, steps and components can be pasted, copied, deleted and resorted. The following operations are possible:

Context menu steps			
✂	Cut	Ctrl+X	Cuts the selected step, including the substeps and the links to components.
📄	Copy	Ctrl+C	Cuts the selected step into the clipboard, including the substeps and the links to components.
📋	Paste	Ctrl+V	Reinserts the step, including the substeps and links to the components, from the clipboard below the selected step.
✕	Delete	Del	Deletes the selected step, including the substeps and the links to components.

	Rename		F2	Allows the selected step to be renamed.
	Update link			
	New Step		Ins	Inserts a new step after the currently selected step and selects this.
	New substep		Shift + Ins	Inserts a new substep in the selected step and select this. If a lower level does not exist, then it is created.
	New step as link...			
	New substep as link...			
	Level Selection Type	multiple (m:n - CheckBox)	Ctrl+T	The steps of a level can either be selected or deselected a multiple number of times (all can be selected/deselected independent of one another) or individually (selecting one deselects all others).
		normal (1:n - RadioButton)	Ctrl+Shift+T	
	Step configuration	Activation		
		Activation disabled		The step activation is only disabled for the selected step
		Activation incl. substeps disabled		The step activation of the selected step, including all of the substeps, is disabled.
		Collapse		
		Exporting into file...		
		Import from file...		
		Load from list		
		Store in list...		
	Go to step...		Ctrl+G	Selects the step with the ID that should be specified.
	Find...		Ctrl+F	
⬆	Upwards		Alt+Up	Shifts the step in the level upwards in the execution sequence.
⬇	Downwards		Alt+Down	Shifts the step at its level downwards in the execution sequence.
	Adapt link...			
	Edit link...			
	Link new components...			Displays the "Component explorer" dialog box, with which a new component can be created. This component is associated with the step.
	Link existing component(s)...			Displays the "Component explorer" dialog box, with which one or more existing components can be linked with the step.
Context menu component				
	Edit Component			Opens the selected component for editing in the component editor or brings it to the foreground if it has already been opened.
✂	Cut		Ctrl+X	Cuts the link to the selected component.
	Copy		Ctrl+C	Copies the association with the selected component to the clipboard.

📋	Paste	Ctrl+V	Pastes the association to the component from the clipboard below the selected component.
✕	Remove link	Del	Deletes the selected component, including the association, only after a confirmation query or only removes its association from the higher-level step.
	Rename	F2	Allows the selected component to be renamed.
↑	Upwards	Alt+Up	Within the step, shifts the component upwards in the execution sequence.
↓	Downwards	Alt+Down	Within the step, shifts the component downwards in the execution sequence.
	Adapt Link		For an external component, it allows the link entry in the component list of the step to be adapted. A new target for the link can be selected in the dialog that opens.
	Edit link...		In a dialog, the link path can be manually changed.
🔍	Search for components and relink...		Opens the "Component explorer" dialog box, with which the selected component can be replaced by another existing component.

Insert a new step

To create a new step, select "New step" via the context menu.

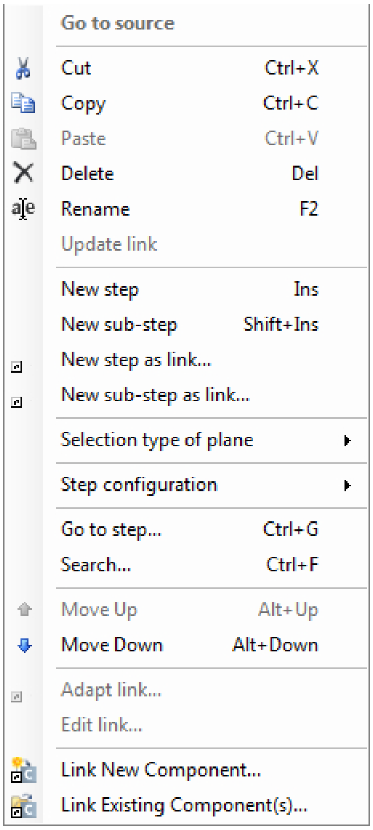


Figure 4-40 Create new step

See also

Dialog: Step tree (Page 155)

4.3.3.4 Linking steps

Introduction

For commissioning machines, sometimes several projects must be managed.

To simplify things for the user, partial or entire step trees from different projects can be linked to one another. As a result of this, recurring step sequences from different projects can be decentrally created and maintained, but centrally managed and executed in one package.

Some examples of this are operator panels, tool magazines, PLC components, etc.

Thanks to this technique of linking the step trees, the commissioning can be reduced to the execution of only one package. The linking of steps allows considerable time savings.

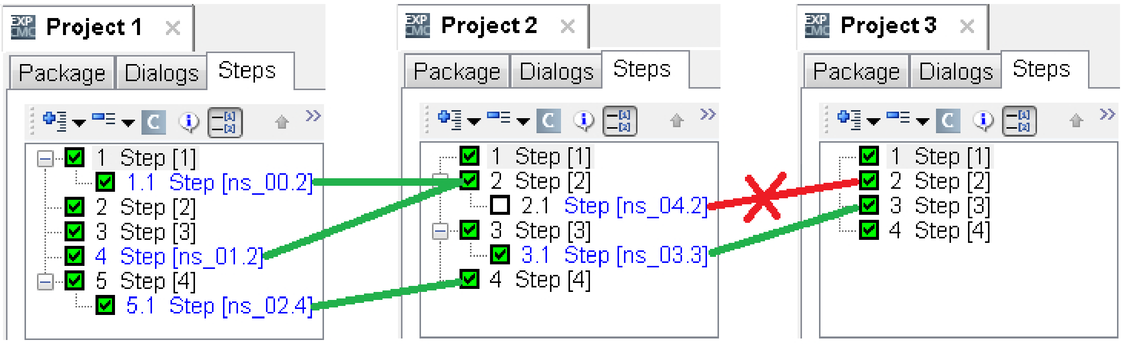
The commissioning engineer sees a complete step tree during project execution. The use of subprojects is no longer visible here.

Due to the linking in step trees, the commissioning of modular machines is supported even more efficiently.

Example

In the following example, the position of a project as the main or subproject is explained and the possible linking depth is shown.

In principle, any project that can run can function as a main project or subproject (an error-free step tree is sufficient in the subproject). A project is designated as a subproject if it serves as a source of the step linking for another project. The main project is thus the project that contains step links from one or more (sub-)projects.



Green connecting line	Permissible step logic operations: • from subproject Project 2 to the main project Project 1 and • from subproject Project 3 to the main project Project 2.
Red struck-through connecting line	**Impermissible** step logic operation: Step [2] of the subproject Project 2 is already used in the main project Project 1. (Linking depth = 1)

The position of the projects as main project or subproject

Project 1 is the main project and can only run with Project 2 (here: subproject). Prerequisite is that the link to Project 3 shown in red does not exist.

Project 2 is both main project and subproject. From the standpoint of Project 1, it is the subproject. As main project, it can run independently with Project 3 (subproject here).

In this example, **Project 3** is the subproject, but it can also run independently (corresponding to a main project).

Rules

To link steps from one project to another project, one project is defined as the main project (MP) and at least one project is defined as the subproject (SP).

The projects remain independent and no project properties are changed. Each project can thus be used as a main project or a subproject.

The following rules must be observed when linking steps:

- In a main project (MP), several links from the same or different subprojects (SP) can be carried out.

  Note

  Linking depth with main projects and subprojects

  The step trees linked from subprojects must not contain any links to another subproject (linking depth = 1).

- Subprojects are separate projects and can run separately even when linked to a main project.
- The steps are linked by the configuration. When the project is deployed, the configured links are dissolved and the linked steps of the subproject (SP) are inserted into the main project (MP).
- The linked steps retain their step ID from the subproject, in which a namespace is added in the main project for each link, in order to ensure unambiguous identification.
  The ID namespace is automatically assigned during configuration. This also prevents an ID namespace from being assigned twice.
  The ID namespace can be changed by the user.
  The linked steps can be recognized in the logbook by a step ID with namespace.
- Steps with namespace before the step ID cannot be linked and cannot contain any linked steps as substeps.
- The execution of the dialogs (in the project folder) is performed exclusively in the main project (MP) also for the steps inserted in the "Steps" tab.
  Only the step tree is linked or taken over from the subproject(s).

- The system variables used in the linked steps of the subproject (SP) always refer to the main project (MP).

> **Note**
>
> **Exceptions are:**
>
> - Up.$Step[ID]
> - Up.$Dialog.StepSelection.CurrentID (see also System variables for the Step tree > Current ID dialog (Page 280))
>
> The system variables can be applied for the main project and subproject.

- The Up script variables are globally applicable in the main project and subproject.
- The same names are permitted for the components in the MP and SP.
- The options of the step configuration ("Activation", "Activation disabled", "Activation disabled including substeps" and "Collapse") can be alternatively managed in the main project or subproject.
  Various use cases are thus supported. A subproject can thus be managed independently of the main project if the "subproject" is selected in the main project under "Manage in".

4.3.3.5 Namespace

Requirement

For each link step which is inserted from a subproject into a main project, a namespace, identified as "ID namespace" is created by default.

A defined namespace only applies in a main project for precisely one link step (blue) with the steps located below it (gray). Thus, the step ID does not have to be changed in the main project or subproject.

Identical step IDs in the main project and subproject are thus possible and also permitted.

Existing queries of the step ID in manipulation tasks and step scripts do not have to be adapted in the main project and subproject.

Query steps

The queries in the main project, which concern a linked step in the subproject, are only expanded by the specification of the namespace.

In the subproject, steps of the main project can be queried by specifying the namespace "." (period).

If the subproject runs separately, the specification of the namespace "." is for a return value "Null". This implementation ensures that a step query always refers clearly to one and the same step, in this case for "." in the main project. This must be taken into account in the programming.

Example for step query

The following shows how a query can be made from one project to another project whether the steps are active. The status in the relation of the projects to one another is either a main project (MP) or a subproject (SP).

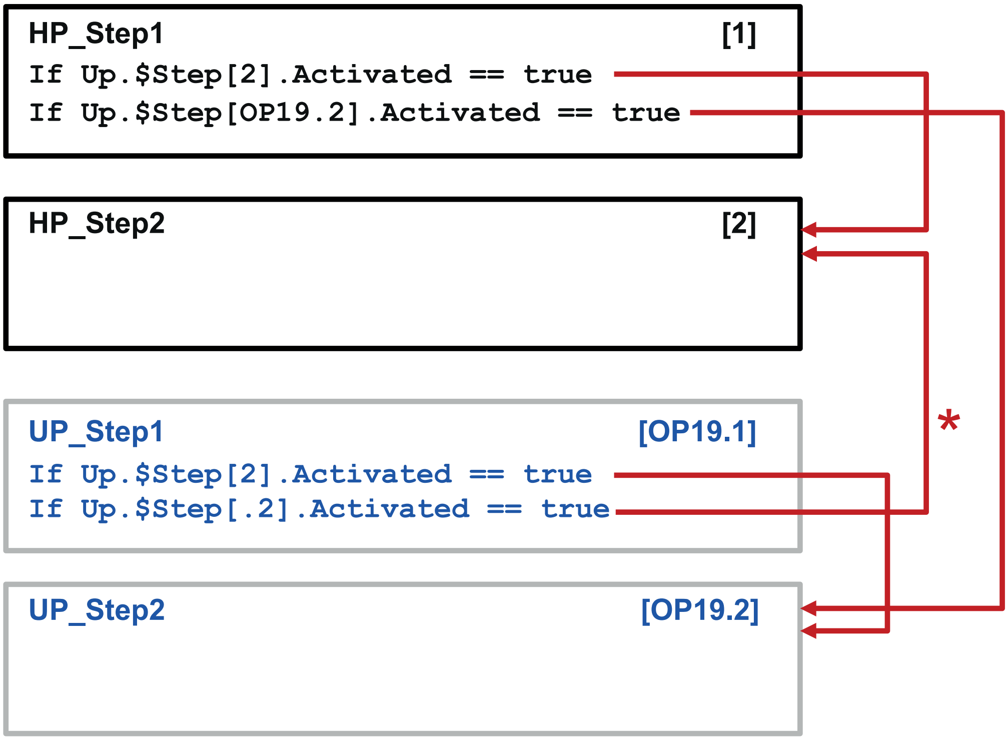

HP_Step1
[1]
If Up.$Step[2].Activated == true
If Up.$Step[OP19.2].Activated == true
HP_Step2
[2]
UP_Step1
[OP19.1]
If Up.$Step[2].Activated == true
If Up.$Step[.2].Activated == true
UP_Step2
[OP19.2]
*

Figure 4-41 Step query - example

* The query in the figure shown above returns a value of "Null" for an independent execution of the subproject (SP).

Before this query, the existence of the step in the main project (MP) can be checked.

Example:

```
If Up.$Step[.2] != Null
```

- OP.19 is namespace in the main project for linked steps from a subproject
- Instead of Activated, the following may apply:

```
If Up.$Step[2].Processing == true
If Up.$Step[2].Locked == true
If Up.$Step[2].Collapsed == true
If Up.$Step[OP19.2].Processing == true
If Up.$Step[OP19.2].Locked == true
If Up.$Step[OP19.2].Collapsed == true
```

Summary example step query

Within a project (the status of the project can be both a main project and/or subproject), the step query is the following:

- `If UP.$Step[2].Activated ==true`

Depending on which direction the query is started in, from MP to SP or vice versa, the condition for the step query must be formulated as follows:

- In the main project to the step of the subproject:
  `If UP.$Step[OP_19.2].Activated == true`
- In the subproject to the step of the main project:
  `If UP.$Step[.2].Activated == true`

Logbook entry

In the logbook, the namespace with a period "." is displayed for steps of the subproject separately from the step ID, e.g. **OP19.1**.

Reserved identifiers

Alphanumerical characters are permissible for the namespace. Reserved identifiers such as UP, Main, True, False, Null are excluded.

Note

Do **not** use any reserved identifiers of the script language, e.g. "True", "False", "Up", "Main", "Null", in the name space.

4.3.3.6 Creating a step link

Procedure

The following shows how a step is copied from a subproject (SP) and inserted into a main project (MP).

1. In the Steps tab in a project of your choice, which you use as a subproject (SP), select a step and copy it via Ctrl+C or the context menu.

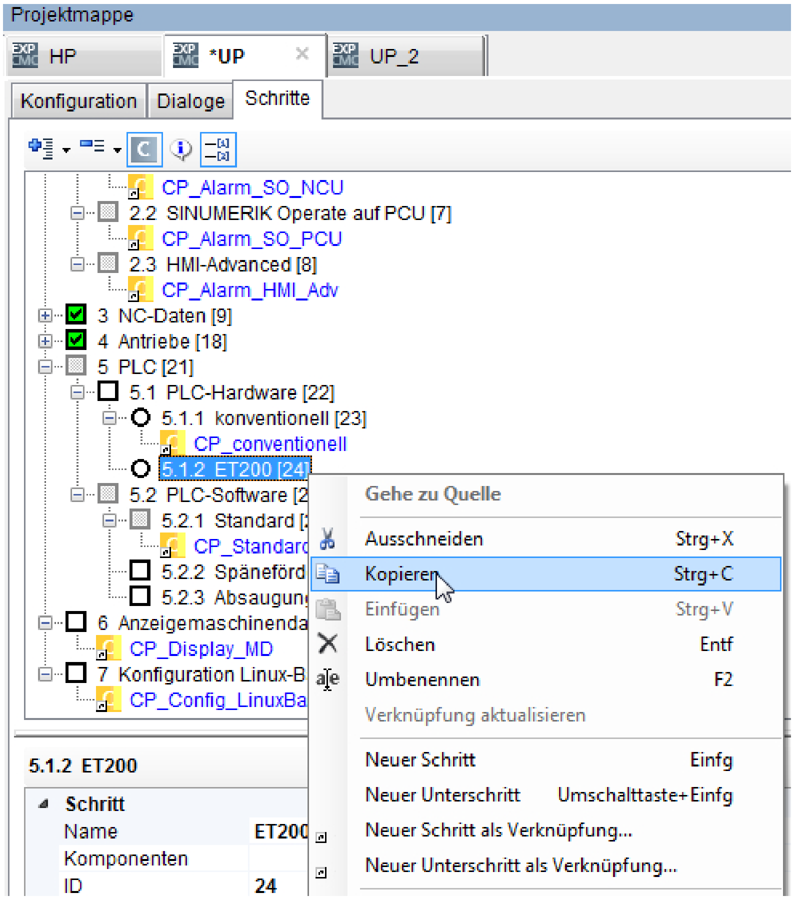


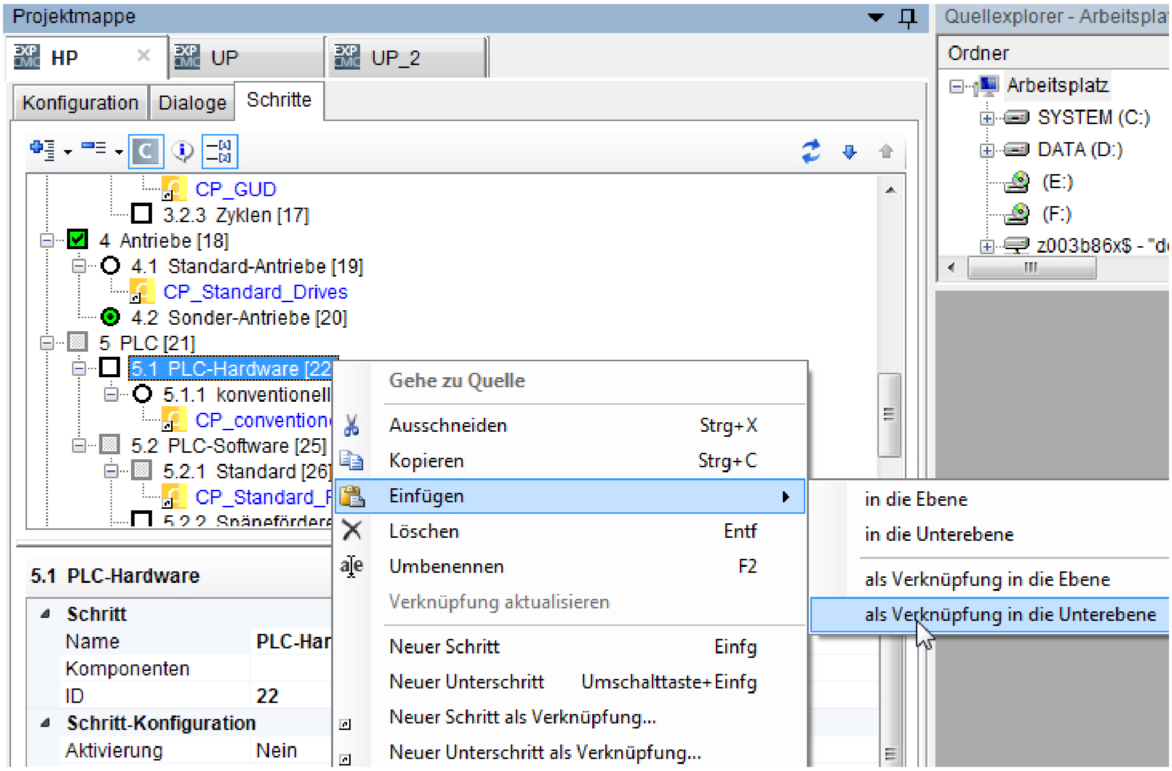
Figure 4-42 Copy link step

Note

Drag and drop is not supported when linking steps.

---

2. Open the main project (MP) and insert the step via Ctrl+V or via the context menu as shown in the following.

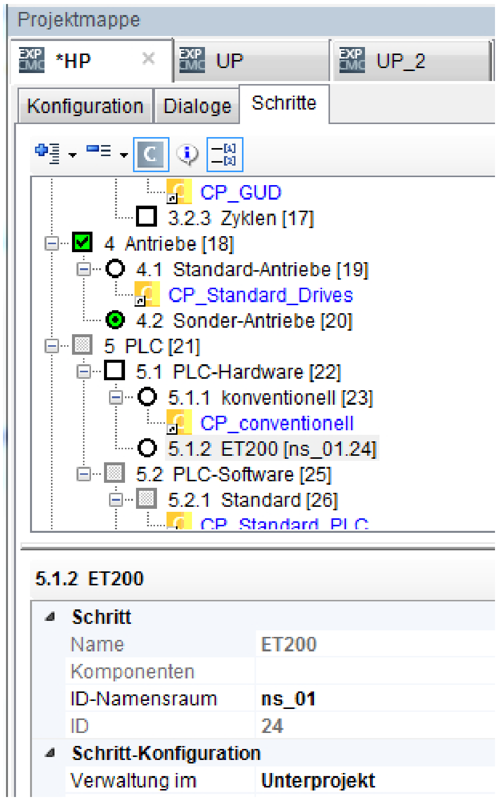


5.1 PLC-Hardware

Schritt
Name PLC-Har
Komponenten
ID 22
Schritt-Konfiguration
Aktivierung Nein

Figure 4-43 Insert link step

3. In the newly inserted step, you can now read that an "ID namespace" has also been assigned in addition to the step ID.
   The "Management in" property is also new; per default the subproject is specified from which the step was copied.



```
CP_GUD
□ 3.2.3 Zyklen [17]
☑ 4 Antriebe [18]
  ○ 4.1 Standard-Antriebe [19]
    CP_Standard_Drives
  ● 4.2 Sonder-Antriebe [20]
▨ 5 PLC [21]
  □ 5.1 PLC-Hardware [22]
    ○ 5.1.1 konventionell [23]
      CP_conventionell
    ○ 5.1.2 ET200 [ns_01.24]
  ▨ 5.2 PLC-Software [25]
    ▨ 5.2.1 Standard [26]
      CP_Standard_PLC
```

5.1.2 ET200

Schritt	
Name	ET200
Komponenten	
ID-Namensraum	**ns_01**
ID	24
Schritt-Konfiguration	
Verwaltung im	**Unterprojekt**

Step properties of linked steps

For the step link and all substeps, all step properties are taken into the main project. The step properties are displayed and saved in the project.

The step properties shown in gray cannot be changed in the main project. The ID namespace produced by the step link can be changed at any time in the link step marked in blue.

The management of the step configuration is set on the subproject when creating a step link. This setting can be changed to the management in the main project.

The "Management in" property specifies whether the step configuration of the gray link step below the blue link step is to be managed in the main project or the in the subproject. The management of the step configuration for the blue link step is always performed in the main project.

4.4 Component editor

4.4.1 Components

So that a component is executed when executing the package, it must be linked with a step and the step must be selected for execution (green symbol). An existing step script can permit or prevent the execution of the components.
The component contains actions. All the relevant operations and data are inserted in the respective actions under the relevant tab during configuration.
The actions to be performed by the component can be activated in the "Actions" tab.

The following actions can be created:

- Execute (Page 184)
- Delete (Page 187)
- Copy (Page 189)
- Change (Page 192)
- Manipulate (Page 194)

During the execution of a component by the package, the activated actions are executed in the order of their listing from top to bottom. If in practice, another order is desired, the order of the actions in the list can be adapted using the toolbar or context menu "Up (Alt+up)" or "Dwon (Alt+down)".

In order to edit an action, it must first be opened by double-clicking or via the context menu. A tab is displayed for this action on which various tasks can be configured.

Files and folders can be inserted below the "Execute" and "Copy" actions both as a copy and a link. A mixture of copied and linked objects is also possible.
In the "Delete", "Change" and "Manipulate" actions, tasks can be directly created using the appropriate operations, imported from a file or linked to an external file. In all cases, it is possible to directly edit the tasks with the task editor integrated in Expert. This has the advantage that the support from Expert is always offered.

All links are only cancelled when the package has been deployed and the data has been accepted in the package. In this case, the advantage of the link is that the source data can be updated independently of the project until deployment.

A version can be assigned to the component. This version is noted in the logbook when executing the package.

Information about the components can be stored in the "Comment" field. The text is not displayed during execution and is also not taken into the logbook.

4.4.2 Execute

Execute action

The "Execute" action allows a program, command or script to be executed. For instance, software components from SIEMENS or other suppliers can be installed or special manipulations made in the archive or other target areas.
The following **target areas** are supported:

- NCU
- PCU (Windows XP) and PCU (Windows 7)
- Linux (runtime system)
- Windows (runtime system)

All the files and folders required for execution can be inserted as a copy or link under the respective target area. When deployed, the links are cancelled and the folders and files are accepted in the component.
Target areas that do not contain any data or operations are shown in gray.

You can specify a command or an executable file in the "Execution of:" text field. It is also possible to transfer arguments, separated by spaces, such as file names, switches, etc. In this case, it must be observed that commands or file names that have a space must be written in inverted commas.
The Special operators (Page 211) can be used at any place in the text field.

Note

The name of the selected file can be transferred to the "Execute" text field using the associated button (↵).

Execute - NCU

When executing, all folders and files are unzipped into a temporary folder. This folder is used as work directory when executing and is then deleted.

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, i.e. a return code not equal to 0 results in an error message and the user must decide whether the package should be cancelled or continued.

See also the section below: "Additional useful notes"

Execute - PCU - Windows XP or Windows 7

When executing, all folders and files on the PCU are unzipped in a temporary folder. When executing, this folder is used as a work directory and is then deleted.
For an installation, additional operator actions may be required on the PCU.
Execution on the PCU is controlled via the PCU handler that runs as a setup application on the PCU.

When a communication error occurs, the PCU can be exited by pressing the key combination Shift+Esc (Shift+ ALARM CANCEL).

Examples of "Execution of"

```
Setup.exe -OF:opfile.sl
Cmd.exe "my start.cmd" /x /y /z
```

PCU handler yields return to Shield

As of Create MyConfig Version V4.7, the following function expansion applies:

The return value of programs which were executed by the PCU handler is returned to the package execution
and logged in the logbook.

Example with return value 0

The return value of the script is evaluated as terminating error code, i.e. a return code not equal to 0 results in an error message and the user must decide whether the package should be canceled or continued.

Execute:

```
cmd /c echo "Examle using exit 0" && exit 0
```

The entry in the logbook that results from this is:

```
Execute
[execute] installations, scripts
Execute PcuWinXP
PcuWinXP
cmd /c echo "Examle using exit 0" && exit 0 started
cmd /c echo "Examle using exit 0" && exit 0 ended 0
```

Example with return value 1

Execute:

```
cmd /c echo "Examle using exit 1" && exit 1
```

The entry in the logbook that results from this is:

```
Execute(1) [execute] installations, scripts
Execute PcuWinXP
PcuWinXP
cmd /c echo "Examle using exit 1" && exit 1 started
cmd /c echo "Examle using exit 1" && exit 1 ended 1
Error
    Error E1109: Execution failed! Command: cmd /c echo
    "Examle using exit 1" && exit 1 Return: 1
```

See also the section below: "Additional useful notes"

Execute - runtime system

A package can be deployed for various runtime systems. In order to now also be able to deploy a package for Linux and for Windows that makes special changes via "Execute" in the "Archive" area, two scripts must be written. Depending on the runtime system, Windows or Linux, the corresponding node is automatically executed and the other skipped during execution.
All folders and files are unzipped in a temporary folder; this folder is then used as work directory and subsequently deleted.

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, i.e. a return code not equal to 0 results in an error message and the user must decide whether the package should be canceled or continued.

Example

All of the files – with the exception of OEM001.SPF – are to 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 when executing, entered as command.

Windows

```
Md .\Save
Move %UP_ARC%\NC\CMA.DIR\OEM001.SPF .\Save\
Del /Q %UP_ARC%\NC\CMA.DIR\*.*
Move .\Save\OEM001.SPF %UP_ARC%\NC\CMA.DIR\
```

Linux

```
mkdir ./Save
mv $UP_ARC/NC/CMA.DIR/OEM001.SPF ./Save/
rm -r $UP_ARC/NC/CMA.DIR/*
mv ./Save/OEM001.SPF $UP_ARC/NC/CMA.DIR/
```

Additional useful notes

Environment variables

`UP_PACK` Location from where the package was started.
Therefore, points to the same path, as "./" - next to the package.

`UP_ARC` Temporary storage location of the archive tree with the subfolders NC, PLC, CP, and DRV

`UP_CFID` ID of the CompactFlash card
Only available for packages with the NCU area.

`UP_SHARE` Folder for exchange release between the PCU and NCU
Only available for Windows packages with the NCU area.
This environment variable points to the shared folder under Windows and to the mount point under Linux.

`UP_PCU_ROOT` Computer name of the PCU
Only available for packages with the PCU area.

`UP_ARC_ROOT`	Folder of the archives to be processed Only available for packages that solely address the ARC area.

General

A script should preferably always be exited with exit 0 if no internal script error handling is performed.

PCU/Windows

Under Windows, the command interpreter should be used as prefix for command line scripts.

For example:

`Cmd.exe /c test.cmd`.

NCU/Linux

With shell scripts (sh) for Linux, it must be ensured that data is saved in the Unix file format, only LF not CR/LF.
The shell ID (e.g. `#!/bin/sh`) must be located in the first line – or the script should be started using a shell command (e.g. `sh my_script.sh`).
A shell should be used as a prefix if you wish to specify commands directly with place holder or environment variables (e.g.: `sh rm /user/*.log`). If a file is specified for execution, './' should be used as a prefix (e.g.: ./test.sh).

4.4.3 Delete

Deleting action

The "Delete" action allows files and folders to be deleted in the target area.
You can create the paths in which the files and folders to be deleted are located on the nodes in the target area. This is supported by the context menu entry "Known folder".

Target areas, that do not contain any delete tasks, are shown in gray, inserted folders that are not permissible, in red.

The following target areas are supported:

Archive	Archive image	NC, MMC, PLC, CP, DRV
NCU	Root of the NCU	card, user, oem, addon
PCU (Windows XP)	Drives/enables of the PCU	C:, D:, E:, F:
PCU (Windows 7)	"	C:, D:
HMI Adv	HMI Advanced drive	DH (data management), USER, OEM, ADD_ON
Operate	Folder of SINUMERIK Operate	hmisl or card

Note

Delete instructions in the target area: Archive/MMC are only applied to the archive and do not affect the data management of HMI Advanced `F:\DH` .

A "Task" must be created using the context menu in order to be able to enter delete operations. The operations can be sub-divided into several tasks in order to improve the structure. The task name can be freely selected.

When executing, the specified files are deleted without any prior confirmation prompt.

Tasks can be created in the following ways:

- New task
  - Enter the operations directly into the operation editor. You can choose any name for the task.
- Importing a task
  - Import the operations that are saved in an external file. The task name is allocated the name of the imported file. After importing, the task name and the operations in the operation editor can be edited without changing the original file.
- Linking a task
  - An external file that contains operations is linked as task. The task name is allocated the name of the linked file and can be subsequently changed. The linked file can still be edited either externally or in the operation editor, whereby the supporting context menu is available in the operation editor.
    When transferring, the task is accepted in the package.

Task editor

The names of the files and folders to be deleted are specified under "Operations". The place holders "`*`" (none or one or several characters) and "`?`" (any character) are permissible in the file names.
On the other hand, *no* place holders are permitted in the folder names. Folders are deleted with all of their contents.
Only one delete operation is permissible per line.
If a line starts with the ";" comment character, it 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.
`0C00022`	The FC22 is deleted.

Additional useful notes

HMI Adv / DH

An error tolerant procedure is used for delete operations as well as when navigating.

A check is first made as to whether the name corresponds to a data management name. If this is the case, then the system uses the information from the data management. Otherwise, a check is made as to whether this name exists at the operating system level and is not assigned to another data management entry. If this is the case, then the system uses the information from the operating system.

Archive/PLC

To delete PLC blocks, these must be specified in the archive name convention.

The naming convention has the format: `TTNNNNN`, whereby `TT` stands for the hexadecimal type, and `NNNNN` for the decimal number of the block. When necessary, both should be filled up with leading zeros to obtain 2 or 5 positions respectively.

PLC block types:

`08 - OB`	Organization block
`0A - DB`	Data block
`0B - SDB`	System data block
`0C - FC`	Function
`0E - FB`	Function block

NCU

In contrast to Windows, the file system used on the NCU differentiates between upper and lower case.

Operate

For the "Operate" target area, create the "All", "NCU" or "PCU" folders using known folders. This folder is used to define for which SINUMERIK Operate the tasks are to be processed.
Under this folder, you define the path to the files to be deleted.
The tasks stored under "All" are processed on the NCU and on the PCU.
The tasks stored under the other two folders are only processed on the associated system.

4.4.4 Copy

Copy action

The "Copy" action enables folders and files in the target area to be copied.
You can insert folders and files in the component as a copy or link under the target areas.

Note

Possible via the context menu "Linking to the folder content"

Link to folder contents means that the link is to a folder, the contents of which are completely copied into the folder of the target area.

The name of the linked folder does not appear in the package.

When you transfer the package, links are canceled and the objects transferred into the component.
The tree view already contains the possible "target areas" as root nodes.
You can create folders below the root nodes using the shortcut menu. The "Known folder" menu item makes entry simpler.
During execution, existing files are overwritten without a confirmation query.

The following color conventions are used to support the operator:

- **gray** - target areas that do not contain any data to be copied
- **red** - folder of file is not permitted in the current path
- **black** - target areas and folders that contain data to be copied

The following target areas are supported:

Archive	Archive image	NC, MMC, NC_MMC, PLC, CP, DRV
NCU	Root of the NCU	card, user, oem, addon
PCU (Windows XP)	Drives/enables of the PCU	C:, D:, E:, F:
PCU (Windows 7)	"	C:, D:
HMI Adv	HMI Advanced drive	DH (data management), USER, OEM, ADD_ON
Operate	Folder of SINUMERIK Operate	hmisl or card

Copying in the data management (HMI Adv, archive)

For folders and files in the data management, names with characters: `A-Z_0-9` and a maximum length of 24 characters are permissible.

Further, especially folders are subject to the data management schematic regarding their names.

The "Known folder" context menu entry makes it easier for you to create folders; this lists all of the permissible subfolders of the selected node.

The following options are available when copying in the data management:

HMI Adv / DH	Copies to the PCU in the F:\DH folder - and at the same time adapts the necessary entries in the HMI Advanced data management (saved in the files: __dhinf.000). This is the preferred procedure in order to transfer large data volumes (bmp, pdf, hlp, ...) to the controller. **NOTE**: When copying files, files with the same name in archive/NC or archive/MMC are deleted.
Archive/NC	Copies to the NC area of the archive. When loading the archive, all the data contained therein is transferred directly to the NC. **NOTE**: When copying files, files with the same name in archive/MMC or HMI Adv / DH are deleted.

Archive/MMC Copies to the MMC area of the archive. When loading the archive, all the data contained therein is transferred directly to the HMI (MMC).

Using the archive/NC_MMC, the functionality of an MMC archive can be implemented to replace data there where it was already located.

NOTE:
When copying files, files with the same name in archive/NC or HMI Adv/DH are deleted.

Archive/NC_MMC Copies to the NC area of the archive, if the file is already available there. Otherwise the file is copied to the MMC area. Regarding loading the archive, the statements made above apply.

NOTE:
When copying files, files with the same name in archive/NC, archive/MMC or HMI Adv/DH are deleted.

The sequence when executing is specified:

```
HMI-Adv/DH → Archive/NC → Archive/MMC → Archive/NC_MMC
```

Additional useful notes

HMI Adv / DH When copying and navigating, an error-tolerant procedure is used. A check is first made as to whether the name corresponds to a data management name. If this is the case, then the system uses the information from the data management. Otherwise, a check is made as to whether this name exists at the operating system level and is not assigned to another data management entry. If this is the case, then the system uses the information from the operating system.

Archive/PLC To copy PLC blocks, they must be specified in the archive naming convention.
The naming convention has the following format: `TTNNNNN`, whereby `TT` stands for the hexadecimal type, and `NNNNN` for the decimal number of the block. When necessary, both should be filled up with leading zeros to obtain 2 or 5 positions respectively.
PLC block types:

`08 - OB`	Organization block
`0A - DB`	Data block
`0B - SDB`	System data block
`0C - FC`	Function
`0E - FB`	Function block

For easier identification of the function block types, the plaintext names in parentheses are shown behind each of the naming conventions.

The blocks must be stored in TTNNNNN format.

NCU — In contrast to Windows, the file system used on the NCU differentiates between upper and lower case.

Operate — For the "Operate" target area, create the "All", "NCU" or "PCU" folders using known folders. This folder is used to define for which SINUMERIK Operate the files are to be copied.
Below these folders, you now create the path for the files to be copied.

The files stored under "All" are stored on the NCU, and when a SINUMERIK Operate is available on the PCU, also saved there. The files stored under the other two folders are only stored on the associated system.

4.4.5 Change

Change action

The "Change" action allows data in the configuration files to be adapted. Configuration files are text files comprising one or more sections including value assignments:

```
[Section]

Identifier=value
```

Generally, the `ini` extension is used for configuration files. However, all files that have the structure of an ini file can be changed.
The possible target areas are specified in the tree view. Using the context menu, additional folders below this can be created, whereby the "Known folder" menu entry simplifies entry.

The following target areas are supported:

Archive	Archive image	NC, MMC, NC_MMC, PLC, CP, DRV
NCU	CompactFlash card of the NCU	user, oem, addon
PCU (Windows XP)	Drives/enables of the PCU	C:, D:, E:, F:
PCU (Windows 7)	"	C:, D:
HMI Adv	HMI-Advanced drive	DH (data management), USER, OEM, ADD_ON
Operate	Folder of SINUMERIK Operate	hmisl or card

To enter change operations, use the context menu to create a "task" in the folder that contains the file to be changed.

The name and path of a task specify the file to be changed.

Tasks can be created in the following ways:

- **New task**
  - Enter the operations directly into the operation editor. You can choose any name for the task.
- **Importing a task**
  - Import the operations that are saved in an external file. The task name is allocated the name of the imported file. After importing, the task name and the operations in the operation editor can be edited without changing the original file.
- **Linking a task**
  - An external file that contains operations is linked as task.
    The task name is assigned the name of the linked file and can be subsequently changed.
    The linked file can be either edited externally or in the operation editor, whereby the supporting context menu is available in the operation editor.
    When deploying, the task is accepted in the package.

Task editor

Based on the structure of the configuration files, change operations should be specified specifying the section. They comprise an identifier, operator and possibly the value. Enter these line-by-line. The context menu supports inserting operators.
Lines that start with a semicolon ";" are not evaluated and can be used for *comments*. The Special operators (Page 211) can be inserted at any place within a line in order to insert the contents of UP variables to be changed.

Note

If folder, file, section or identifier are not available, then the package itself generates these in order to implement a value assignment.

Operators

=	Assignment of a new value
~	Deleting data, i.e. identifier and value are removed
?=	Assigning a (new) value, if no (old) value is available

Example:

```
[LANGUAGE]
FontList=Europe, Europe, Europe, Europe, Europe
[Local]
ADRESS0=0,NAME=/NC,LINE=11
ADRESS1~
[Function]
Del. Dir query1?=111
```

Additional useful notes

Operate

For the "Operate" target area, create the "All", "NCU" or "PCU" folders using known folders.
This folder defines for which SINUMERIK Operate the tasks are to be processed.

Under this folder, you define the path to the files to be modified.

The tasks stored under "All" are processed on the NCU and on the PCU. The tasks stored under the other two folders are only processed on the associated system.

Changes of display data of the SINUMERIK Operate are possible via the actions "Change" and "Manipulate" in CMC. It makes sense to use only one of the two methods (preferably "Manipulate"), because they affect one another.

4.4.6 Manipulate

Manipulate action

The "Manipulate" action allows machine data to be accessed, essentially independent of the data storage in the control system.
The data to be manipulated can be stored in the archive (e.g. NC data, drive data) as well as in the HMI (display data).

Data manipulation operations are supported in the following "target areas":

- NC data
- SINAMICS data
- Display data

Additional information on the possibilities associated with manipulation tasks is provided in the description of the script language (see: Introduction to scripting (Page 197)).

A "Task" must be generated using the context menu in order to be able to enter manipulation operations. The operations can be sub-divided into several tasks in order to improve the structure. The task names can be freely selected.

Tasks can be created in the following ways:

- **New task**
  - Enter the operations directly into the operation editor. You can choose any name for the task.
- **Importing a task**
  - Import the operations that are saved in an external file. The task name is allocated the name of the imported file. After importing, the task name and the operations in the operation editor can be edited without changing the original file.
- **Linking a task**
  - An external file that contains operations is linked as task.
    The task name is assigned the name of the linked file and can be subsequently changed.
    The linked file can be either edited externally or in the operation editor, whereby the supporting context menu is available in the operation editor.
    When deploying, the task is accepted in the package.

Task editor

Manipulation tasks are executed by the package and **not** by the NC or the HMI.
Processing is line-oriented up to the end of the line or the comment character '`;`'.
Except for string values, upper/lower-case and spaces are not relevant for the notation.
Example: "`ID = 0`" corresponds to "`id=0`"

Note

Exception

String values form an exception as upper/lower-case and spaces are relevant.

In a simple case, data manipulation operations comprise consecutive assignments: Block number (optional), identifier, operator and value. A section should be set first, such as the channel number or the drive object path.

The context menu provides support when inserting sections, control structures, functions, procedures, operators and system variables. A catalog is also available listing all the identifiers specified by SIEMENS.

The operation editor marks the syntax of faulty entries with a red wavy line. Syntax errors in the task can be easily found using the "Go to next error" context menu entry.

The formatting (simplified Qt-html) within your texts for the functions and procedures can be checked via the "Message preview" context menu entry. To do this, mark the entire text within the quotation marks and click "Show preview" in the context menu.

Example of an operation

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

Additional useful notes

NC data

NC data is manipulated, based on the NC archive area.

If a context is not set using CHANDATA(n), then channel 1 is assumed.

NOTICE:
With multi-channel assignments, make sure that CHANDATA(1) is set in front of data that is **not** channel-dependent!

If NC data is to be manipulated, in relation or depending on its old value (e.g.: operators `+=`, `*=` etc.), then it must be ensured that this data is also in the NC offline archive (take into account machine data: 11210 UPLOAD_MD_CHANGES_ONLY and 11212 UPLOAD_CHANGES_ONLY).

In the case of "Source of the data," "Initial state," or "Overallreset," all NC data is always provided.

When loading an archive to organize the memory, the NC requires certain machine data that is in the *upload files* (NC.OPT, GLOBAL.IN, SCALING.INI and CHAN.INI) located in front of INITIAL.INI. Before manipulation, the package loads all data from the INITIAL.INI to the memory, completes the manipulation operations and then writes back all the data. All previously existing data is written back to its old position, both in the INITIAL.INI and in the upload files. Additional data is attached to the data of the INITIAL.INI - and according to the following rule - also to the data of the upload files:

- NC.OPT, if the identifier starts with $ON
- GLOBAL.INI, if the identifier starts with $MN or $MA
- SCALING.INI, if the identifier starts with $MN or $MA
- CHAN.INI, if the identifier starts with $MN or $MC

If NC data is manipulated, then the *line checksum* is generated and attached to all data.

SINAMICS data

Drive data is manipulated, based on the DRV archive area.

Display data

Display machine data is saved differently for HMI-Advanced and SINUMERIK Operate.

For *HMI-Advanced*, data is read from the section: `[OPI settings]` of the file: "MMC.INI" in the search sequence: `HMI_ADV → MMC2 → ADD_ON → OEM → USER` but is only written back to `MMC.INI` under `USER`. When deleting data, this results in a reset to a user-specified value of preceding files in the search sequence.

For *SINUMERIK Operate*, data is read from the section: `[OPI settings]` of the file: "hmi_md.ini" in the search sequence: `siemens → addon → oem → user`, but is only written back to `hmi_md.ini` under `user`. When deleting data, this results in a reset to a user-specified value of preceding files in the search sequence.

4.5 Scripting

4.5.1 Introduction to scripting

In the Scripting section you will find general information on script instructions:

- Color coding of program elements (Page 199)
- Set channel (Page 199)
- Cross-area access (read) (Page 200)
- Package variables (Page 201)
- Comments (Page 204)

Call scripting

Scripting an be used in:

- Dialogs
- Steps
- Manipulation jobs

Further information on scripting and "Inserting a script" can be found in the section Scripts under Dialogs, Steps, and Manipulation jobs (Page 204).

Dialogs

On selected dialog pages, scripts can be executed before displaying the pages, when clicking the "Next" button, and at the end. The processing of the current and subsequent dialogs can be controlled with these scripts.

Proceed as follows to call scripting in dialogs:

1. In the Dialogs tab, choose Dialog scripts > Scripts.
2. Use the ... button to open the ScriptEditor dialog.
3. Right-click in the dialog to select via the Insert Script context menu.
4. Select from the available options:
   - Section
   - Identifier
   - Check structure
   - Operator
   - Function
   - Procedure

Steps

Under every step in the step tree, scripts can be executed before the step is processed. The current and all subsequent steps can be controlled from these scripts.

Proceed as follows to call scripting in the Steps tab:

1. Choose Step scripts > Scripts.
2. Click ... to open the ScriptEditor Step dialog.
3. Right-click in the dialog to select via the Insert Script context menu.
4. Select from the available options (see dialogs).

Manipulation jobs

In the Component Editor, scripts can be added under actions of the type "Manipulate".

Proceed as follows to use scripting for manipulation jobs:

1. Select the Manipulate tab in the Component editor.
2. In the Instructions window, right-click to open the Insert Script context menu.
3. Here you can select between available options (see dialogs).

Section Scripting describes a simple programming language based on known elements of other programming languages. Scripts can be created for dialogs, steps, and manipulation tasks.

The task editor supports creation of scripts with syntax checking, syntax emphasis, automatic completion, and with an extensive shortcut menu.

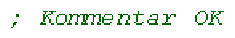
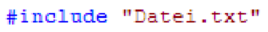
Overview of the script language

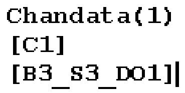
Fundamentals		**Control structures**	
↓ Set channel (Page 199)	CHANDATA(1)	↓ if control structure / if-else control structure (Page 206)	If ...
↓ Sections (Page 204)	[B3_S3_PS1]	↓ while control structure (Page 208)	While ...
↓ Area data (Page 206)	R[1], p105	↓ Logic operators / comparison operators (Page 211)	true != false
Package variable		**Procedures and functions**	
↓ Package variables (Page 201)	Up.name	↓ Procedures (Page 231)	Msg(...)
↓ Script variables (Page 202)	Up.myVar	↓ Exists (Page 229)	Input(...)
↓ DO variables (Page 203)	Up.doVar.doNr	**Miscellaneous**	
↓ Step activation	Up.$Step[1]	↓ Commenting (Page 204)	; ...
↓ Querying package properties (Page 272)	Up.$Pack	-	-
↓ Querying dialog properties	Up.$Dialog	-	-
Assignment and calculation		↓ Special operators (Page 211)	$(Up.ax)
↓ Assignment operators (Page 209)	= ~ ?= ...	-	-
↓ Arithmetic operators (Page 210)	+ - * / & |	-	-

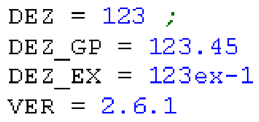
4.5.2 Color coding of program elements

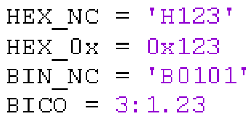
For better orientation, important program elements are color coded in the task editor and in the step condition editor. The remaining program elements are displayed black and unformatted.

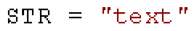
The meaning of the colors is shown in the following:

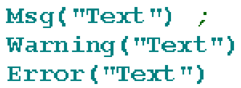
Program element	Color	Formatting	Example
Comment	Dark green	Italic	 `; Kommentar OK`
Keyword	Blue	Bold	 `#include "Datei.txt"`

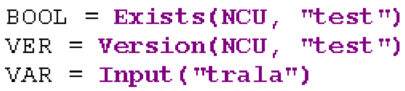
Sections	Black	Bold	
			`Chandata(1)`
			`[C1]`
			`[B3_S3_DO1]`

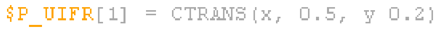
Decimal numbers	Blue	-	
			`DEZ = 123 ;`
			`DEZ_GP = 123.45`
			`DEZ_EX = 123ex-1`
			`VER = 2.6.1`

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

String	Chestnut brown	-	 `STR = "text"`

Procedures	Dark cyan	Bold	
			`Msg("Text") ;`
			`Warning("Text")`
			`Error("Text")`

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

User frames (variables that have not been interpreted)	Yellow/orange	-	 `$P_UIFR[1] = CTRANS(x, 0.5, y 0.2)`

4.5.3 Set channel

Set channel - **CHANDATA(?)**

For NC data with the same name that comes a multiple number of times (e.g. channel data) before value assignments, the channel to which it belongs must be specified. This channel specification is effective as long as no other channel setting is made via CHANDATA() operation. Data accesses between areas are unaffected by this.

When the channel is not specified, then channel 1 is automatically used.

Examples

CHANDATA(1)

```
;channel 1 is set as an area as of 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
```

CHANDATA(2)

```
;Channel 2 is set as of here
$MC_CHAN_NAME = "Channel 2"
```

[C3] `;alternative abbreviation`

```
;Channel 3 is set as of here
```

NC data

Syntax	Description
NC[Ci].<NC-variable>	The value of the NC machine data "NC variable" in the "i" channel within the NC archive. If the data is not found, the null value is returned. The channel specification "i" must be assigned a value between 1 and 10. **Example**: NC[C5].$MC_CHAN_NAME

4.5.4 Cross-area access (read)

As an alternative to setting the channel for NC data, the channel can also be specified directly during access (local). This channel specification has a higher priority compared to the last setting for the channel (`CHANDATA()` operation).

As an alternative to setting a section, the section can also be specified directly during access (local). This section specification has a higher priority compared to the last setting for the section (section setting `[B?_S?_PS??],[ADV], [SL]`).

In order to be able to access channels or sections from step scripts, cross-area access must be used.

Syntax

The syntax for cross-area access is generally:

```
Namespace[section].variable
```

Namespaces

The following namespaces are defined:

Namespace / target area	Channel / section input	Format example

BD (display data)	HMI of the target system
	● `ADV - HMI-Advanced`
	● `SL - SINUMERIK Operate`
		`BD[ADV].$MM_LCD`
		`BD[SL].$MM_LCD`

NC (NC data)	Channel C1, C2 … C10	`NC[C2]. $MC_MM_NUM_AC_PARAM`
PS (SINAMICS data)	Subpath within the drive system B3_S3_PS3, B3_S15_PS99 …	`PS[B3_S3_PS1].p105`

Example: Cross-area access in step script

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

Example: Cross-area access in the NC data manipulation task

```
[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
```

4.5.5 Package variables

Package variables - `Up.name`

Package variables are variables that only have meaning within the package. The controller does not know these variables and they are not transferred to the controller.

Variable	Syntax	Description
Script variable	Up.<variable>	Value of a UP variable
DO variable	Up.<variable>.<property>	Property value a DO variable
System variable	Up.<variable>	Value of a system variable

Package variables are identified by the prefix "Up.". Depending on their definition location, package variables are classified according to the following categories:

Category	Defined by	Value allocation by
Script	User in manipulation tasks	User
DO (drive object)	User on the "SINAMICS Topology" dialog page	System
System	The system using the package configuration	System

All package variables are defined in the same namespace. Consequently unique names must be used throughout all categories. Package variables are addressed using the prefix `Up`. They can be used throughout the package, in step scripts, tasks, etc.

They can be used as follows:

- In manipulation tasks for reading as well as writing using `Up.name`
- In step-scripts (see Steps (Page 166)) reading via `Up.name`
- In connection with the replacement operator: `$(Up.name)`(see Special operators (Page 211))

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`. Upper/lower-case is not relevant. Package variables that receive their value from the system can only be accessed in read-only mode.

4.5.5.1 Script variables

Script variables - `Up.myVar`

Users create script variables using operations with the form:

```
  Up.name = value
```

using a unique name and immediate value initialization.

Examples

```
Up.M1=100.123
Up.xx_1= R[7]
Up.NX_BOOL=true
Up.channel="channel
_1"
R[5] = Up.M1
```

4.5.5.2 DO variables

DO variables - `Up.doVar`

The DO variables (drive object variables) that can be used in the package are managed in the package design on the " Dialog: SINAMICS topology (Page 143)." This DO variable list can be exported as an XML file and linked again. To achieve data consistency, the same DO variable list can be associated with a topology project in Topo as a file. The variables are instantiated and the property values allocated when executing on this dialog by assigning to the existing drive components.

In manipulation tasks, DO variable properties can be read-accessed via:

```
  Up.Variable.Property
```

.

The following properties are available for a DO variable with the name `doVar`:

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 drive object

Note:
A 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

Up.doVar.dpSlot

Supplies the PROFIBUS location number, starting with 1 (only for SERVO of the NCU and NX).

4.5.5.3 System variables

Note

When the password variables are used, the password that is used appears in the logbook during the package execution.

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 name ("`sysname`") preceded by a '`$`'.

System variables of the following type exist:

- `Up.$Pack` - System variables for the Package tab (Page 271)
- `Up.$Dialog` - System variables for the Dialogs tab (Page 273)
- `Up.$Step[ID]` - Steps tab (Page 282)
- `UP.$Env` - Environment variables (Page 289)

Note

You can find a reference list on the system variables in the last section of Create MyConfig - Expert under Reference list system variables (Page 271).

4.5.6 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.

In order to comment out several lines simultaneously, select the corresponding lines and use Ctrl+K+C.

To undo commenting out, use Ctrl+K+U.

Example

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

4.5.7 Scripts under dialogs, steps, and manipulation jobs

4.5.7.1 Sections

Set section - `[B?_S?_PS?], [ADV], [SL]`

For data with the same name, which occur in various files, before assigning a value, a specifying section must be specified. This section is effective as long as no other section is specified.

Depending on the target area, the following applies:

Target area	Explanation	Format
SINAMICS data	Subpath within the drive system	`[B3_S3_PS3]` **Notice** **No leading zeroes must be used** No leading zeroes must be used for the figures. These are **not** detected during the validation run for replacement operators.

Display data	HMI of the target system	`[ADV] - HMI Advanced`
		`[SL] - SINUMERIK Operate`

Examples

```
[B3_S3_PS3]                        ; SINAMICS drive data
p1082[1] = 9000                    ; Maximum speed
[SL]                               ; SINUMERIK Operate display data
$MM_FIRST_LANGUAGE = 1
```

Display data

Syntax	Description
BD[<section>].<BD variable>	Value of the display machine data "BD variable" from the "section" section. Values for "section": • ADV for HMI Advanced • SL for SINUMERIK Operate If the data is not found, null is returned. **Note**: EMB - HMI Embedded is no longer supported. **Example**: BD[ADV].$MM_TECHNOLOGY

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 number "psNr" Values from 1 to 99 are valid for "psNr". Leading zeros are not permitted! **Example**: PS[B3_S3_PS1].p105

4.5.7.2 Area data

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

All variables/identifiers without the package prefix `Up.` are interpreted as area data. They can be written (left-hand side of an assignment) as well as also read (right-hand side of an assignment).

Note

Cross-area variable access is only supported for read access. In order to transfer the value of a data item from one area to another area, this can be assigned in the target area, whereby values can be read from different source areas (see Cross-area access (read) (Page 200)).

4.5.7.3 Check structures

Control structures - context menu

The following variants of the control structures are available:

- if control structure (Page 206)
- if-else control structure (Page 206)
- if-elsIf control structure (Page 207)
- if-elsIf-else control structure (Page 207)
- while control structure (Page 208)

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 operation blocks dependent on whether the "exp" condition is fulfilled or not. If this is fulfilled (true), then the first operation block "block1" is executed, otherwise, the second execution "block2".

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

Note

Before using machine or drive data in control structures, it is necessary to specify a section.

Example

```
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
```

if-elsIf control structure / if-elsIf-else control structure

The IF-ELSE control structure was expanded in Create MyConfig V4.7. After the IF condition, alternative conditions can be checked by specifying ELSIF or ELIF.

The operation block is executed after the first fulfilled condition.

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 operation blocks dependent on whether the "exp" condition is fulfilled or not.

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

Once a condition or alternative condition is fulfilled, the associated operation 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 operation 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 operation block.

Each operation block can, in turn, contain control structures; this means that 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
```

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
```

4.5.7.4 Operators

Overview operators

The following operator groups are available for scripting in dialogs, step scripts, and manipulation jobs:

- Assignment operators (Page 209)
- Arithmetic operators (Page 210)
- Logic operators (Page 211)
- Comparison operators (Page 211)
- Special operators (Page 211)

Assignment operators

Assignment operators - `VAR = 123`

The assignment operators constitute the core function of manipulation jobs, because control data is written with them. All assignment operators can also be applied to script variables.

The following assignment operators are available:

?=	Assign if blank or not available Assign the specified value if no value is available.
=	Assign Assignment of a new value An assignment of "null" leads to the deletion of the variables or to the reset of the value to the default, see deletion operator ~.
~	Delete Delete data, i.e. reset to the default value
+=	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 of the specified and existing value (setting of bits).
&=	AND operation AND operation of the specified and existing value (resetting of bits).
:=	Assignment without evaluation Assignment without evaluation to the right of the operator. The value is transferred to the controller as specified.

If no old value exists for compiled operators: `+= -= *= /= &= |=`, then an error message is output.

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)
```

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 4-11 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 a clear 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_MM_TOOL_MANAGEMENT_MASK = $MN_MM_TOOL_MANAGEMENT_MASK |
'B10000'
```

Logic operators / comparison operators

Condition- `true != false`

The following logical operations and comparison operators are available to form Boolean expressions:

Table 4-12 Logic operators

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

Table 4-13 Comparison operators

==	equal to
!=	not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

Logic operations must be set in brackets in order to define a clear sequence.

Example

```
(UP.$Step[HMI_Adv]&&Version(PCU, "HMI-Advanced")>=7.6)||
(Version(NCU, "InternalVersion")!=2.7)
```

Special operators

Special operator Replace - `$(up.?)`

With the aid of the special operator **Replace**, it is possible to replace parts of an operation, an identifier, a value or a (message) text flexibly with the value of an `Up.name` variable.

The following rules must be observed when using the special operator Replace:

- It can only be used with package variables (see Package variables (Page 201)).
- During replacement, inverted commas are removed from string values.
- It can be nested.

The special operator Replace can be used in the following:

- in sections, e.g.: `[$(up.doX.pspath)]` or `[B3_S$(Up.slave)_PS$(Up.doNr)]`
- in operations under Manipulate and Change, e.g.: `$MA_MAX_AX_VELO[AX$(UP.AXNr)]=1000`
- in control structures under Manipulate, e.g.: `If p107[$(Up.i)] == 254`
- in step scripts, e.g.: `Version(PCU, "HMI Advanced")>=$(Up.HMI)`
- in functions and procedures under Manipulate, e.g.: `MSG("Machine number: $(Up.Masch_Nr)")`
- in command lines for "Execution of:", e.g.: `Setup.exe > Log_$(Up.Masch_Nr).txt`
- in message texts of the steps, e.g.: "The machine number is: `$(Up.Masch_Nr)`"

Note

Instruction lines in which a special operator Replace is used cannot be checked for correct syntax in a verification run.

4.5.7.5 Functions

Note

Please note that variables are permissible as arguments for functions.

Functions for dialogs, steps, and manipulation jobs

The following functions can be used in scripts for dialogs, steps, and manipulation jobs.

The functions are called via the context menu Insert Script > Function.

- Input(..) (Page 218)
- InputChoice(..) (Page 219)
- InputEnum(..) (Page 223)
- InputText(..) (Page 224)
- InputInt(..) (Page 225)
- InputUInt(..) (Page 226)
- InputReal(..) (Page 227)
- Round(..) (Page 228)
- Match(..) (Page 228)

- Exists(..) (Page 229)
- Version(..) (Page 229)
- DateTime(..) (Page 230)

Input functions

Two types of Input functions are available for expert:

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

Input functions linked to the 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-bound input functions are assigned a user-specified value. The user-specified value can be entered directly or only applied at runtime via variables and replacement operators.

Independent of the selection "Display message", for the "Step tree" dialog, all Input functions are displayed.

Note

Double quotation marks

If the message text <msg> or another string should contain double quotation marks, they must be written with single quotation marks (' " ') so that they are not interpreted by CMC.

Example: Msg("Component ' " ' HMI Install ' " ' ...")

This procedure must be used for all double quotation marks within strings, just as is typically seen in the NC.

Note

Inserting line breaks

Line breaks can also be inserted in texts using '<br>'.

Example:

Input("line <br> new line")

In tasks, a preview dialog box can be called to check the formatting after selection of the line or the text via shortcut menu item "Message preview."

Extended programming notes on input functions

Programming options

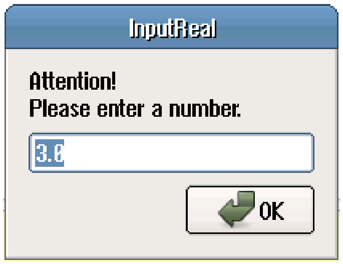
Examples are listed below that you can program for the input functions.

This is demonstrated with the InputReal function as an example. However, you can use any other input function.

Example of the representation of a line break

For a clearer arrangement, insert a line break between the header and the text.

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

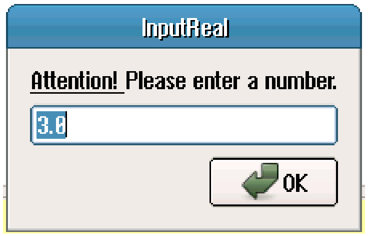


Attention!
Please enter a number.
3.0

Figure 4-44 Example of the representation of a line break

Example of the representation of a header with underscore

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

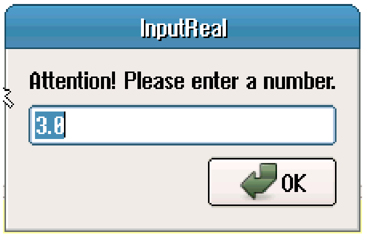


Attention! Please enter a number.
3.0

Figure 4-45 Example of a header with underscore

Example of the representation of a bold font

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



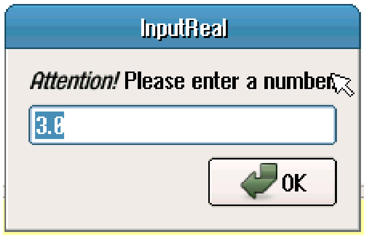
Attention! Please enter a number.

3.0

Figure 4-46 Example of a header, bold

Example of the representation of an italic header

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



Attention! Please enter a number.

3.0

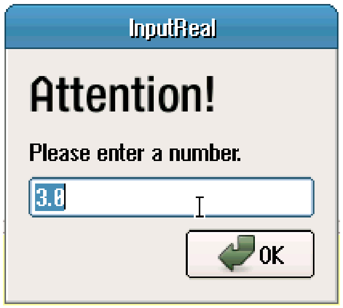
Figure 4-47 Example of a header, italic

Example of the representation of a larger header

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

Example of a header, large

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



Attention!

Please enter a number.

3.0

Figure 4-48 Example of a large header in a separate line

Example of a header in another font

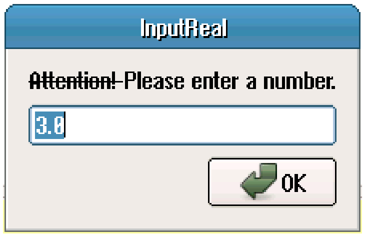
```
up.type=InputReal("<tt> Notice! </tt> Please enter a number.",3.0)
```

Example of a header, small

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

Example of a header, strikethrough

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



~~Attention!~~ Please enter a number.

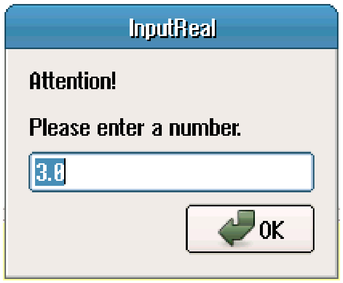
3.0

Figure 4-49 Example of the representation of a header, strikethrough

Example of an empty line between the header and the text

In order to display a warning, for example, a better representation can be achieved via an empty line.

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



Attention!

Please enter a number.

3.0

Figure 4-50 Example of empty lines between the header and the text

Example of a header in superscript

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

Example of a header in subscript

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

Examples of colored headers

Some examples of colored texts are shown below.

Red header

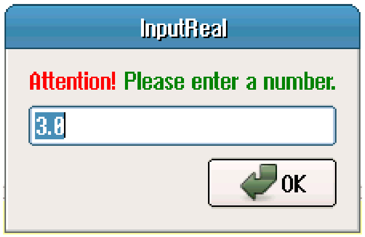
```
up.red=InputReal("<b style='color:red'> Notice! </b
style='color:red'> Please enter a number.",3.0)
```

Green header

```
up.green=InputReal("<b style='color:green'> Notice! </b
style='color:green'> Please enter a number.",3.0)
```

Header and text in different colors

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



InputReal

Attention! Please enter a number.

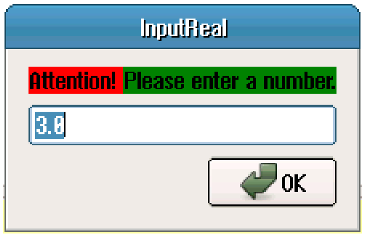
3.0

Figure 4-51 Example of the representation of the header and text in different colors

Example of background color

Header and text have different background colors, e.g. red and green

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



InputReal

Attention! Please enter a number.

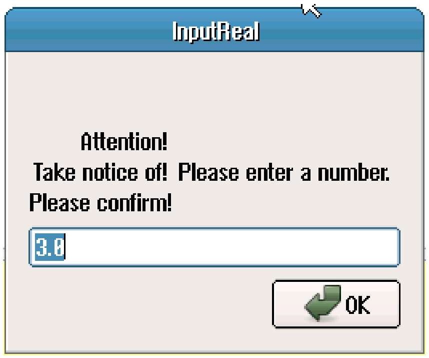
3.0

Figure 4-52 Example of the representation of text with a background color

Example of an invisible table

You can layout your texts using and invisible table.

```
up.table=InputReal("<br><table><tr><td align='right'> Notice! </td>
<td> Please note!!! </td> </tr><tr><td align='right'>Your data will
be lost </td><td> (please perform a backup </td></tr></table><br>
please confirm!",3.0)
```



Attention!
Take notice of! Please enter a number.
Please confirm!

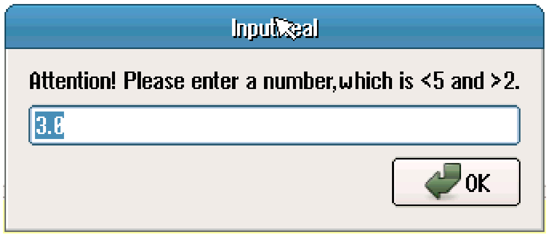
Figure 4-53 Example of an invisible table

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! Please enter a number that is &lt;5
and &gt;2.",3.0)
```



Attention! Please enter a number,which is <5 and >2.

Figure 4-54 Example of special characters

More detailed information on HTML code can be found in the relevant documents and on the Internet, e.g. at HTML code (http://www.html-seminar.de/referenz-sonderzeichen.htm).

Input

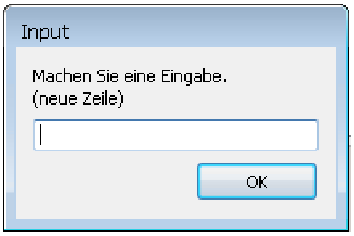
Input("<msg>")

The Input function is a non-type-specific function.

Using the `Input("<msg>")` function, the operator can query a value while the script is being executed.
If the operator does not specify a value, the function returns `null`. This can then be intercepted in the script code using an appropriate IF condition.

Example input

```
Up.in = null
While Up.in == null
 Up.in = Input("Make an entry. <br>(new line)")
EndWhile
```



Machen Sie eine Eingabe.
(neue Zeile)

Figure 4-55 Example input

InputChoice

If the function `InputChoice(<icon_label>,<buttons>)` is called, a dialog is shown with an icon during package execution in which the user can choose directly by pressing individual buttons.

The selection made is returned as a string.

InputChoice(<icon_label>,<buttons>)

The entry can be described using the "`icon_label`" argument. An icon can optionally be chosen at the beginning of the argument "`icon_label`", separated by a semicolon, which is displayed during package processing.

You can choose between the following icons "`icon_label`":

- Selection without icon
- *I; selection with information icon
- *Q; selection with question mark icon (InputChoice icon)
- *W; selection with warning icon
- *E; selection with error icon

The "`buttons`" argument can be used to specify the identifiers for multiple buttons, separated by semicolons. The freely selectable identifiers simultaneously define the possible return values of the function.

Control characters in the argument "`buttons`":

- * 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 a identifier.

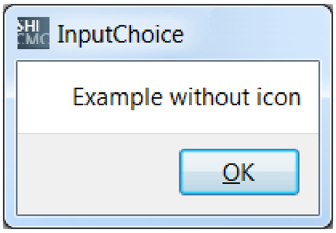
The control character '&' in the identifier for buttons defines the subsequent letters as "Alt-shortcut" (key combination Alt+<letter>).

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

The Alt-shortcuts are not case-sensitive.

Example Selection without icon

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

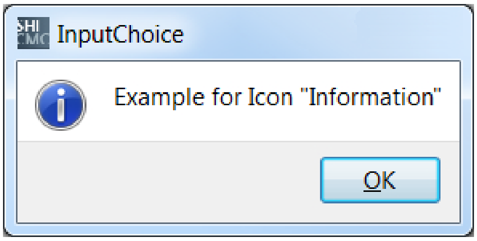


Example without icon

Figure 4-56 Example without icon

Example Selection with information icon

```
Up.retI = InputChoice("*I;Beispiel für Symbol '"'Information'"'",
"*&OK")
Up.retI = InputChoice("*I;Example for Icon '"'Information'"'",
"*&OK")
```

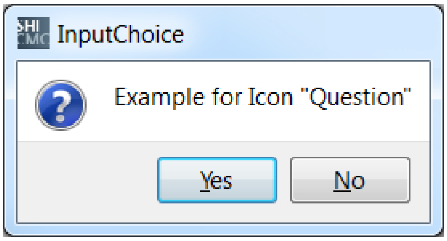


Example for Icon "Information"

Figure 4-57 Example for "Information" icon

Example Selection with question mark icon

```
Up.retQ = InputChoice("*Q;Beispiel für Symbol '"'Frage'"'
(InputChoice)", "*&Ja;&Nein")
Up.retQ = InputChoice("*Q;Example for Icon '"'InputChoice'"'",
"*&Yes;&No")
```

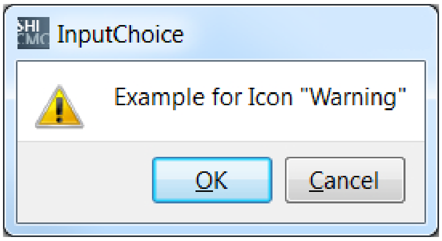


Example for Icon "Question"

Figure 4-58 Example for "InputChoice" icon

Example Selection with warning icon

```
Up.retW = InputChoice("*W;Beispiel für Symbol '"'Warnung'"'",
"*&OK;&Abbruch")
Up.retW = InputChoice("*W;Example for Icon '"'Warning'"'",
"*&OK;&Cancel")
```

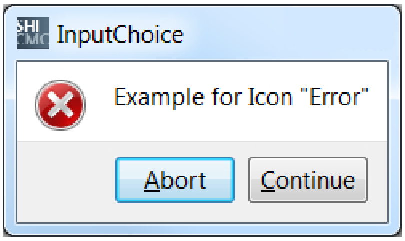


Example for Icon "Warning"

Figure 4-59 Example for "Warning" icon

Example Selection with error icon

```
Up.retE = InputChoice("*E;Beispiel für Symbol '"'Fehler'"' (error)",
"*&Abbruch;&Fortsetzen")
Up.retE = InputChoice("*E;Example for Icon '"'Error'"'",
"*&Abort;&Continue")
```



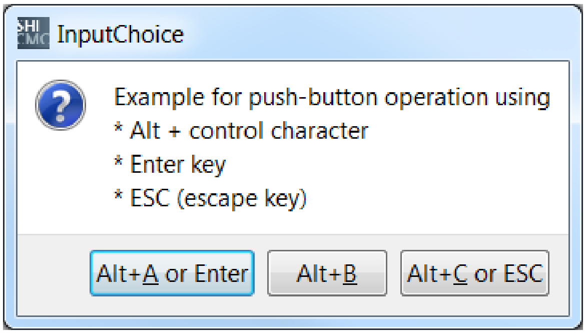
InputChoice

Example for Icon "Error"

Figure 4-60 Example for "Error" icon

Example Pushbutton operation

```
Up.example = InputChoice("*Q;Beispiel für Schaltflächenbedienung
mit<br> * Alt-Tastenkombination<br> * Eingabe<br> * ESC", "*Alt+&A
oder Eingabe;Alt+&B;^Alt+&C oder ESC")
Up.example = InputChoice("*Q;Example for push-button operation
using<br> * Alt + control character<br> * Enter key<br> * ESC (escape
key)", "*Alt+&A or Enter;Alt+&B;^Alt+&C or ESC")
```



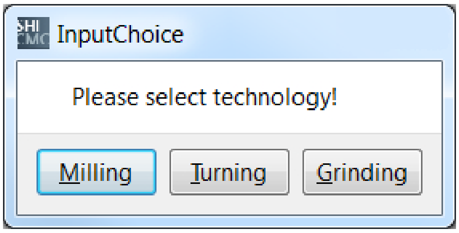
InputChoice

Example for push-button operation using
* Alt + control character
* Enter key
* ESC (escape key)

Figure 4-61 Example for Pushbutton operation

Example Technology selection

```
Up.MaschineType = InputChoice("Wählen Sie die Technologie!",
"&Fräsen; &Drehen; &Schleifen")
Up.MaschineType = InputChoice("Please select technology!",
"&Milling; &Turning; &Grinding")
```



InputChoice

Please select technology!

Figure 4-62 Example "Technology selection"

InputEnum

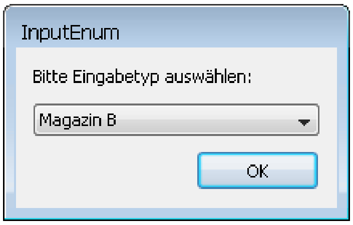
InputEnum(label,enums)

When the `InputEnum(label,enums)` function is called, when the package is executed, a dialog box is displayed and the operator can make a selection.
This chosen string is assigned to an Up variable.

- The entry can be described using the "`label`" argument.
- You can define the selection text sequence using the "`enums`" argument, separated with semicolon.
- You can define this as preselection with a "*" at the beginning of a selection text sequence. If this is not done, 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")
```



InputEnum

Bitte Eingabetyp auswählen:

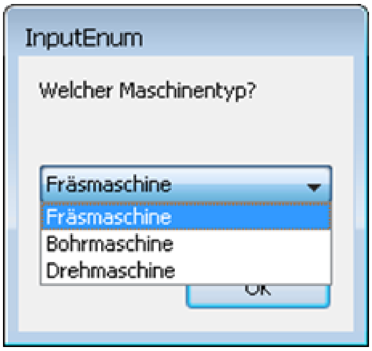
Magazin B

Figure 4-63 Example with defined preselection

Example Preselection with replacement operator

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

```
Up.Typ1 = "Milling machine"
Up.Typ2 = "Drilling machine"
Up.Auswahl = "$(up.Type1); $(Up.Type2); turning machine"
Up.Maschinentyp = InputEnum("Which machine type?", "$(Up.Auswahl)")
```



InputEnum

Welcher Maschinentyp?

Fräsmaschine

Figure 4-64 Example Preselection with replacement operator

InputText

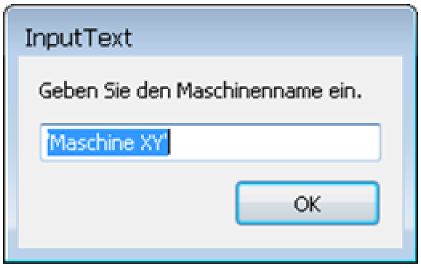
InputText(label,text)

Mit Aufruf der Funktion `InputText(label,text)` wird bei der Paketabarbeitung eine Dialogbox angezeigt und der Bediener kann einen Text eingeben.
Dieser String wird einer Up-Variablen zugewiesen.

- The entry can be described using the "label" argument.
- A text can be preselected using the "text" argument.
- The dialog box is changed over to hidden password entry using the "*" user-specified value.
- The return value is the text that the user had entered.
- For an empty input field and when pressing "OK", an empty string " " is returned.

Example Input text with and without replacement operators

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



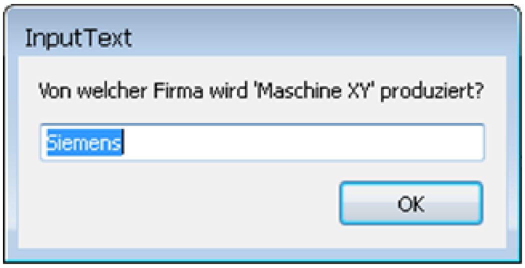
InputText

Geben Sie den Maschinenname ein.

'Maschine XY'

Figure 4-65 Example Input text without replacement operator

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



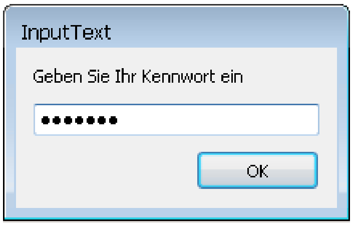
Von welcher Firma wird 'Maschine XY' produziert?

Siemens

Figure 4-66 Example of input text with replacement operator

Example of input text for password entry

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



Geben Sie Ihr Kennwort ein

•••••••

Figure 4-67 Example of using input text for password entry

InputInt

InputInt(label,int)

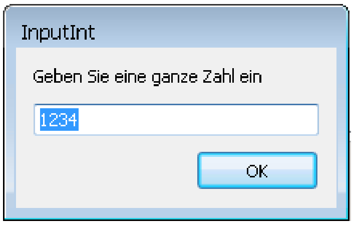
By calling the `InputInt(label,int)` function, a dialog is shown during package processing in which the user can enter an integer.

This integer value is assigned to an Up variable.

- The entry can be described using the "label" argument.
- The "int" argument must contain a default value. This can be changed while the package is being executed.
- The return value is the number that the user had entered.

Example of entering an integer

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



InputInt

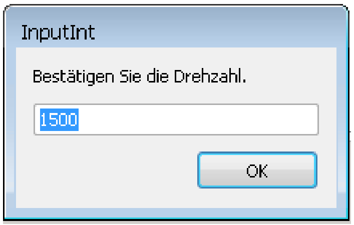
Geben Sie eine ganze Zahl ein

1234

Figure 4-68 Example of entering an integer

Example of replacement from string, e.g. from an Enum

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



InputInt

Bestätigen Sie die Drehzahl.

1500

Figure 4-69 Example of replacement from string, e.g. Enum (enumeration)

InputUInt

InputUInt(label,uint)

By calling the `InputInt(label,uint)` function, a dialog is shown during package processing in which the user must enter an unsigned integer.

This unsigned integer value is assigned to an Up variable.

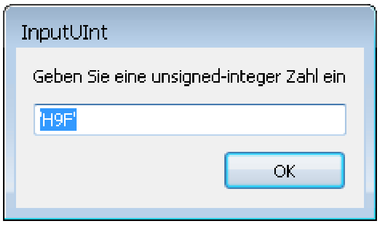
- The value can be entered in various formats.

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

- The entry can be described using the "`label`" argument.
- The "`uint`" argument must contain a default value. This can be changed while the package is being executed.
- The return value is the number that the user had entered.

Example

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



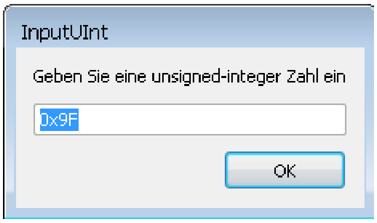
Geben Sie eine unsigned-integer Zahl ein

'H9F'

Figure 4-70 Example of InputUInt entry unsigned integer number

Example of specification with replacement operator

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



Geben Sie eine unsigned-integer Zahl ein

0x9F

Figure 4-71 Example of specification with replacement operator

InputReal

InputReal(label,real)

When the `InputReal(label,real)` function is called, when the package is executed, a dialog box is displayed and the operator can enter a real number.
This real string is assigned to an Up variable.

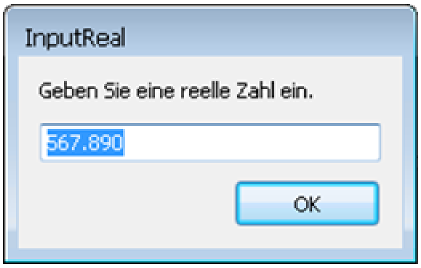
- The entry can be described using the "label" argument.
- The "real" argument must contain a default value. This can be changed while the package is being executed. The following entries are permitted:

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



InputReal

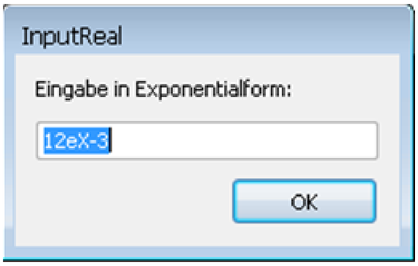
Geben Sie eine reelle Zahl ein.

567.890

Figure 4-72 Example of real-number entry

Example of specification with variables

```
Up.Exp = 12eX-3    ; = 0.012
Up.RealZahl = InputReal("Input as exponential:", Up.Exp)
```



InputReal

Eingabe in Exponentialform:

12eX-3

Figure 4-73 Example of specification with variables

Round

```
Round(<Value>, <Precision>)
         Up.pVerst = round(0.12345, 3) -> 0.123
         Up.pVerst = round(p1460[0], 2) -> 0.62
```

Match

Match

The Match function searches in the `<string>` character string according to the search pattern `<regular expression>`.

Regular expressions are used as search pattern.

```
Match(<string>, <regular expression>)
```

Example

```
Up.NCU = match("NCU720.3 PN", "NCU([0-9]+)") -> "720"
Up.NCU = match(Up.$Env.NCU, "NCU([0-9.]+)") -> "720.3"
```

Exists

Exists() function - `Exists(<area>, "<path>")`

The `Exists(<area>, "<path>")` function returns "true" if the folder or the "path" file exists with the absolute path in the "area" area.

Note

Only absolute paths are permitted, relative paths cause errors in the verification run.

The following values are permitted for "area":

- NCU - for access to the file system of the NCU (e.g. the CompactFlash card /card/)
  Example: EXISTS(NCU, "/siemens/versions.xml").
  Upper/lower-case must be taken into account in the path. Please note that only areas can be addressed, which have also been activated in the project folder when configuring the package.
- PCU - for access to the hard disk of the PCU 50, the path must start with the drive letter, e.g. "D:\" or "D$\"
  Example: EXISTS(PCU, "F:\DH")
- ARC - for access to data in the archive image, the path must start with NC\, MMC\, PLC\, CP\ or DRV\
  Example: Exists(ARC, "NC\CST.DIR\Test.SPF")

Example Output of a message and entry in the log file

Output of a message and entry in the log file if the "/siemens/versions.xml" file has been found on the NCU.

```
IF EXISTS(NCU, "/siemens/versions.xml")
  Msg("The versions.xml file exists on the NCU.")
ENDIF
```

Example Package variable `UP.DHexists`

Package variable "UP.DHexists" has the value "true" if the data management "F: \DH" has been found on the PCU.

```
Up.DHexists = EXISTS(PCU, "F:\DH")
```

Version() function - `Version(<area>, "app")`

The `Version(<area>, "app")` function supplies the version of an application "app". Please note that only versions that have also been activated in the project folder when configuring the package can be queried in the areas. For example, when querying versions on the NCU, then the NCU area must be activated, etc.

If the version of the specified "<app>" application can be determined, then the function returns the `null` value. In this way, it is possible to react when a version cannot be found.

For the package area <area> = "PCU", all of the SIEMENS applications installed on the PCU are determined, which are entered in the PCU registry under the key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\AUTSW
```

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

For the name of the application "app", an abbreviated, unique entry 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.

Example of query as to whether a version is larger/smaller

Query whether the version of the NCK is less than or equal to 83.03:

```
Version(NCU, "NCK") <= 83.03
```

Version of HMI Advanced greater than or equal to 6 and ShopMill is installed:

```
Version(PCU, "HMI-Advanced") >= 6 && Version(PCU, "ShopMill") != null
```

For the entry SINUMERIK Operate: 04.07.02.00 from Version.xml

Version of HMI Advanced greater than or equal to 4.7 and ShopMill is installed:

```
Version(PCU, "SINUMERIK Operate") >= 04.07 && Version(PCU,
"ShopMill") != null
```

For instance, the queries can be used in control structures (If, While).

DateTime

DateTime

If the function DateTime("") is transferred with an empty string "", it is formatted according to ISO 8601 "yyyy-MM-ddThh:mm:ss".

The following script code must be used to get the previous name of the logbook.
Script_201412031651

```
Up.dt = DateTime("yyyyMMddhhmm")
Up.$Dialog.PackageEnd.LogName = "$(Up.Pack.Name)_$(Up.dt)" ; extension
by Shield correspond to password protection
Up.$Dialog.PackageEnd.LogDir = "./" ; is default ==> only explicitly
written down once more
```

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 examles in
combination
```

4.5.7.6 Procedures

Various procedures exist in order to be able to interactively influence script editing. These should be coded in a dedicated line.

Table 4-14 Overview of procedures:

Procedure	Description
`Msg("?")`	Message output, which must be confirmed with "OK." <msg> contains an arbitrary message text that is also written to the logbook.
`Error("?")`	Message output, which must be confirmed with "OK" and results in termination of the package. <msg> contains an arbitrary message text that is also written to the logbook.
`Warning("?")`	Message output and prompt for cancellation of the package. <msg> contains an arbitrary message text that is also written to the logbook.
`Log("?")`	Logbook entry <msg> contains an arbitrary message text that is only written to the logbook.
`Logging(On|Off)`	Logging on/off Logging in the logbook can be activated and reactivated.
`ExtCall("<path>")`	External call Call of an external manipulation task. There is no verification run for this task in Expert, and therefore no syntax check. The file must be stored in ANSI code. The path can be specified as an absolute path or relative path. 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.

Procedure	Description
`#Include "<path>"`	Deployment directive Preprocessor operation that includes an external file (manipulation task) when checking or deploying. ● When a package is deployed, the contents of the external file are taken over in the manipulation task instead of the #include operation. ● Included files are also checked by the verification run. ● The path can be specified as an absolute path or relative path. With internal tasks, the relative path refers to the storage folder of the component. With external tasks, the relative path refers to the storage folder of the external task. The path can contain environment variables of the type %SUB_PATH%.
`Return()`	Function Return() This function can be used in the script of the dialog pages. With **OnInit**, after the dialog page is displayed and processed, a switch to the next dialog page takes place. With **OnNext**, the dialog page is processed. With **OnEnd**, the dialog page is terminated.
`Skip()`	Function Skip() This function can be used in the script of the dialog pages. A dialog page is skipped with the function Skip(). **Exception**: The dialog pages Archive access, NCU access, PCU access and System configuration are not skipped if "Yes" is set in the **Project / Package** tab under **Package > Data areas > Use archive**. An error message is displayed during package processing in Shield or during verification in Expert.
`Redo()`	Function Redo() This function can be used in the script of the dialog pages. With **OnInit**, an error message is displayed in Expert during the verification run. With **OnNext**, a return is made to the operator input of the dialog page. With **OnEnd**, only after successful execution is the script executed at the end of the dialog page.

Note

Double quotation marks

If the message text <msg> or another string should contain double quotation marks, they must be written with single quotation marks (' " ') so that they are not interpreted by CMC.

Example: Msg("Component ' " ' HMI Install ' " ' ...")

This procedure must be used for all double quotation marks within strings, just as is typically seen in the NC.

Note

Inserting line breaks

Line breaks can also be inserted in the <msg> texts via '<br>'.

Example:

Msg("line <br> new line")

In tasks, a preview dialog box can be called to check the formatting after selection of the line or the text via shortcut menu item "Message preview".

Example

```
If Up.doX == null
 Warning("X drive was not assigned!")
Else
 Log("assignment, X drive checked.")
EndIf
```

4.6 Conversion

Projects that were created with the versions SCI 2.6, CMC 4.4 / 4.5 or CMC 4.6 are automatically converted on calling.
Packages that were created with these versions cannot be converted during direct import.

Pay attention to the procedure for converting packages from older versions of Expert.

Procedure

1. Import the packages with the corresponding older version of Export.
2. Open and convert the project that has been created using the actual CMC Expert.
3. Store the project and forward it as package with the current version (*.exe, *uaz, *usz).

4.6.1 Rules for adapting the syntax

If you open a project SinuCom Installer SCI 2.6, CMC V4.4 / 4.5 or CMC V4.6 with the current version, the majority of the syntax from scripts and step scripts is converted automatically. However, there are a few expressions where an automatic conversion is not possible. These must by adapted manually to the correct syntax.

This section provides an overview of which expressions can be converted automatically and which must be manually adapted. Syntax errors (syntax that has become invalid) are displayed through the verification of package by Expert. Especially extensive conversions are necessary in projects of SCI 2.6.

Adaptation of logic operators in projects from Version 2.6

Syntax Version 2.6	Conversion possible?	Syntax as of Version 4.6:
&	Yes	&&
|	Yes	||

Adaptation of values in projects from Version 2.6

Replace the **UP()** function with a value query.

Syntax Version 2.6	Conversion possible?	Syntax as of Version 4.6:
Up($Step[id])	Yes	Up.$Step[id]
UP(data)	Yes	UP.data
UP(doX) != null & UP(doX.slaveNr) == 3	Yes	UP.doX != null && UP.doX.slaveNr == 3

Adaptation of functions in projects from Version 2.6

Adapt the **Version()** function. Enter the range to determine the version.

Syntax Version 2.6	Conversion possible?	Syntax as of Version 4.6:
Version(NCK) <= 4.4.18	Only in part: **Example**: Version(NCK) → Version(?NCU|PCU, "NCK")	Version(NCU, "NCK") <= 4.4.18
Version(HMI Advanced) >= 6	Only in part: **Example**: Version(HMI Advanced) → Version(?NCU|PCU, "HMI Advanced")	Version(PCU, "HMI Advanced") >= 6
Version(ShopMill) != null	Only in part: **Example**: Version(ShopMill) → Version(?NCU|PCU, "ShopMill")	Version(PCU, "ShopMill") != null
Exists(<area>, <path>)	Only in part: Set in "", but only the ARC, NCU, PCU areas are still possible.	Exists(<area>, "<path>")

Replace the **NC()** function with a value query.

Syntax Version 2.6	Conversion possible?	Syntax as of Version 4.6:
NC($MC_CHAN_NAME, 5) == "CHAN5"	Yes **Special case**: NC($MC_CHAN_NAME) → NC[C1].$MC_CHAN_NAME	NC[C5].$MC_CHAN_NAME == "CHAN5"

Replace the **BD()** function with a value query.

Syntax Version 2.6	Conversion possible?	Syntax as of Version 4.6:
`BD($MM_TECHNOLOGY, ADV) == 2`	Only in part: Set in "", but only the ADV, SL sections are still possible. (EMB not applicable)	`BD[ADV].$MM_TECHNOLOGY == 2`

Querying SINAMICS drive parameters

Syntax Version 2.6	Conversion possible?	Syntax as of Version 4.6:
Not directly possible	-	PS[B3_S3_PS1].p105

Sections in tasks of SINAMICS data

The sections in tasks of SINAMICS data are converted from the long notation still tolerated in SCI into a new shorthand notation, if the section does not contain a replacement operator.

Syntax Version 2.6	Conversion possible?	Syntax as of Version 4.6:
[BUS3.DIR/SLAVE3.DIR/ PS000099.ACX]	Partially: Only if a replacement operator is not contained.	[B3_S3_PS99]

4.7 Transferring a project/creating a package

Transmission of a CMC package to a third party

Note

Legal information on Open Source Software and Commercial off-the-shelf

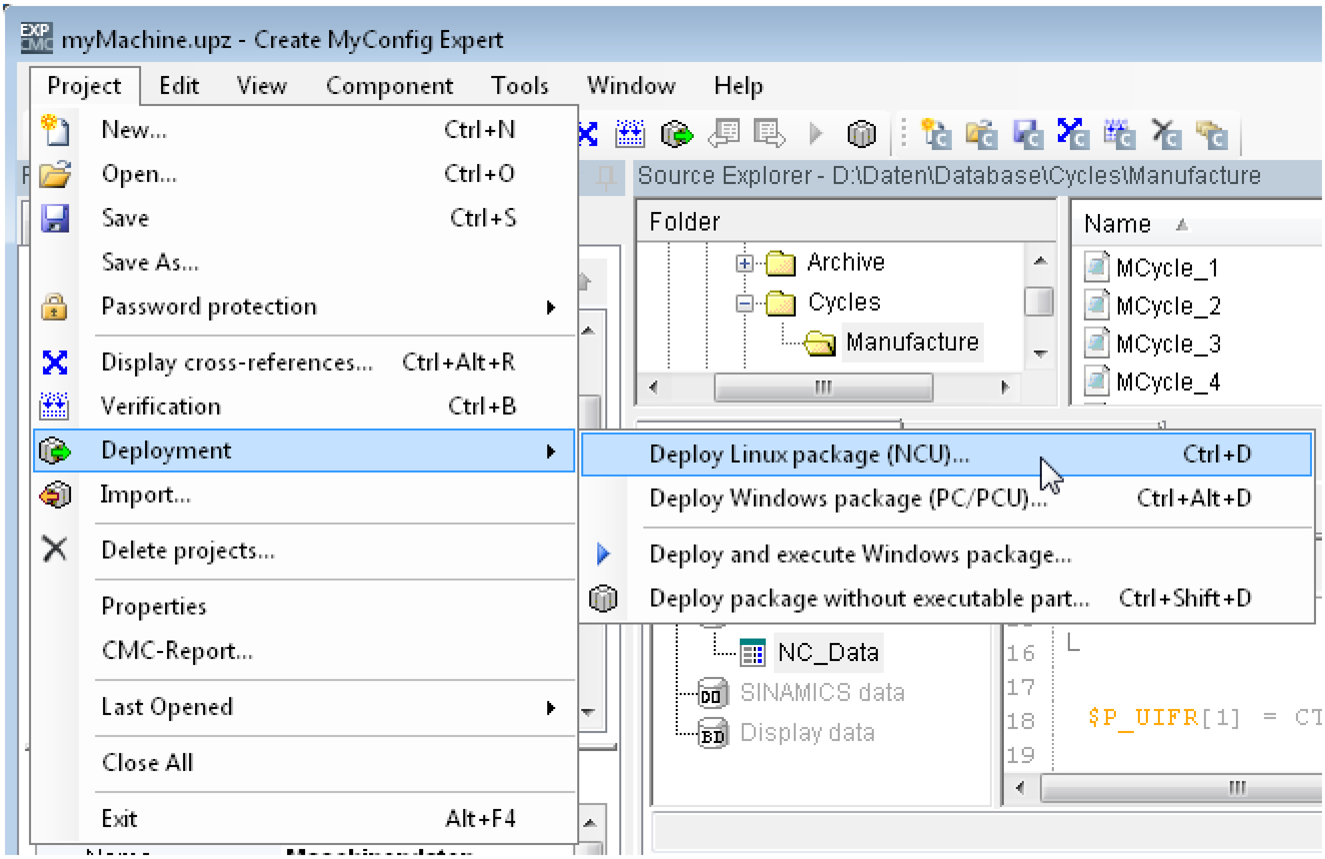
If a CMC package in which Shield with Open Source Software or Commercial off-the-shelf is integrated, is transmitted to a third party, the CMC package must contain explicit information on the legal conditions.

In order not to violate the licensing agreement, include the "Notes for conformation" dialog page in the file on the CD. This file is located in the root directory of the CD.

Procedure

The generated Expert projects can be deployed as Windows or Linux packages. These can then be executed under the particular operating system.

To create a package, proceed as follows:



1. Select "Project > Deployment" and select whether you want to create a Windows package or a Linux package.
2. Select a name and a target directory for the package.
   An automatic verification is started which checks the package for inconsistencies and errors. You can also perform the verification manually via the "Verification run" button.

4.8 Cross references

4.8.1 Overview

Cross reference objects

Projects generated in CMC Expert can be very complex, and generally contain a large number of cross-reference objects.

Cross-reference objects are identifiers of machine data, user and system variables, linked files and folders. These cross-reference objects are configured at various positions of use within the project and the linked components. In the form of a table, the cross-reference function shows you where all of the cross-reference objects are used.

This table can be exported using the context menu.

4.8.2 Display cross references

Display options

There are several ways of displaying cross references. Cross references can be displayed for a project with all linked components, for an individual component or for a cross-reference object (e.g. an identifier in a manipulation task).

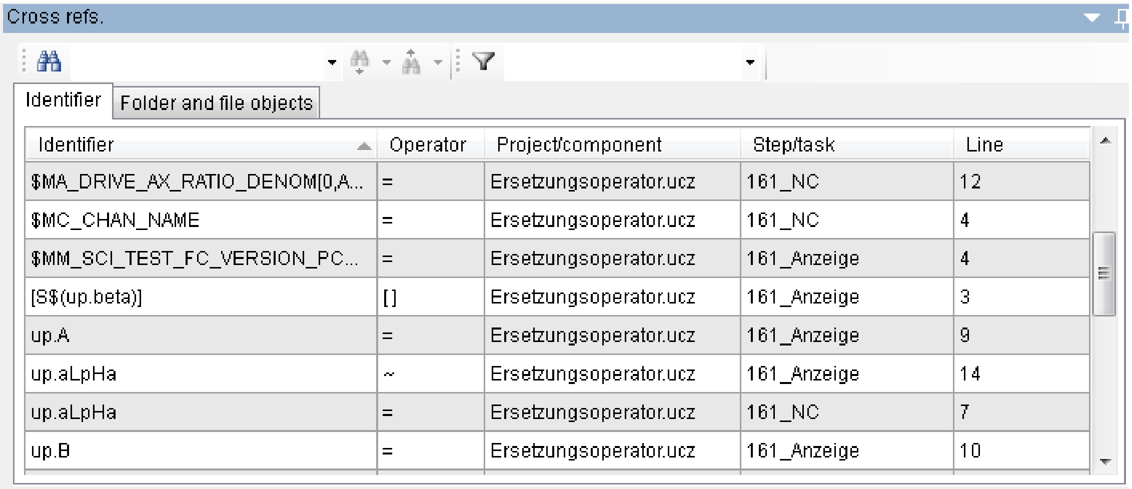
For the project with all linked components, select "Project > Display cross-references" and for individual components "Component > Display cross-references."
Optionally you can display the corresponding cross references using the symbol bar.

To display individual cross-reference objects, put the cursor on the object in the editor and select "Go to position of use" from the shortcut menu. The cross-references are then filtered according to the object name. The object name of part of it can also be entered in the filter field manually.

Overview of the user interface

The list of locations where cross reference objects are used is grouped using the two "Identifier" and "Folder and file objects" tabs:

"Identifier" tab



Identifier	Operator	Project/component	Step/task	Line
$MA_DRIVE_AX_RATIO_DENOM[0,A...	=	Ersetzungsoperator.ucz	161_NC	12
$MC_CHAN_NAME	=	Ersetzungsoperator.ucz	161_NC	4
$MM_SCI_TEST_FC_VERSION_PC...	=	Ersetzungsoperator.ucz	161_Anzeige	4
[S$(up.beta)]	[]	Ersetzungsoperator.ucz	161_Anzeige	3
up.A	=	Ersetzungsoperator.ucz	161_Anzeige	9
up.aLpHa	~	Ersetzungsoperator.ucz	161_Anzeige	14
up.aLpHa	=	Ersetzungsoperator.ucz	161_NC	7
up.B	=	Ersetzungsoperator.ucz	161_Anzeige	10

Identifier Identifier is the unique identification of the cross reference object within the list.

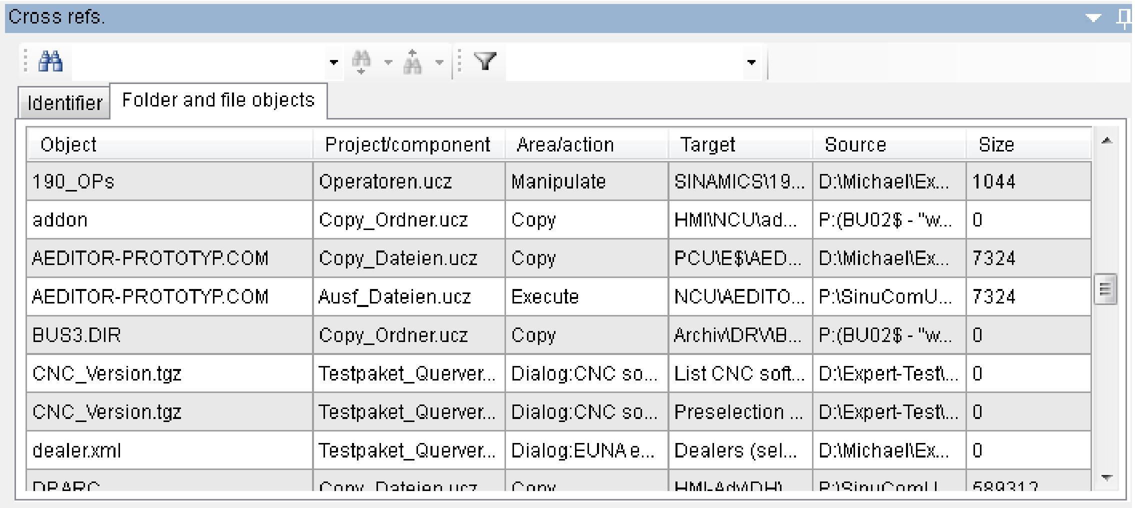
Operator The operator specifies how the cross reference object is processed at the position where it is used.

Project/ Component Indicates in which project or in which component the cross-reference object is located.

Step/ task Indicates in which step or task the cross-reference object is located.

Line Indicates in which line, the cross-reference object is located.

"Folder and file object" tab



Object	Project/component	Area/action	Target	Source	Size
190_OPs	Operatoren.ucz	Manipulate	SINAMICS\19...	D:\Michael\Ex...	1044
addon	Copy_Ordner.ucz	Copy	HMI\NCU\ad...	P:(BU02$ - "w...	0
AEDITOR-PROTOTYP.COM	Copy_Dateien.ucz	Copy	PCU\E$\AED...	D:\Michael\Ex...	7324
AEDITOR-PROTOTYP.COM	Ausf_Dateien.ucz	Execute	NCU\AEDITO...	P:\SinuComU...	7324
BUS3.DIR	Copy_Ordner.ucz	Copy	Archiv\DRV\B...	P:(BU02$ - "w...	0
CNC_Version.tgz	Testpaket_Querver...	Dialog:CNC so...	List CNC soft...	D:\Expert-Test\...	0
CNC_Version.tgz	Testpaket_Querver...	Dialog:CNC so...	Preselection ...	D:\Expert-Test\...	0
dealer.xml	Testpaket_Querver...	Dialog:EUNA e...	Dealers (sel...	D:\Michael\Ex...	0
DPARC	Copy_Dateien.ucz	Copy	HMI-Adv\DH\	P:\SinuComU	589312

Object Indicates which file or which folder is involved.

Project/component Indicates in which project or in which component the cross-reference object is located.

Area/action Indicates in which area of the project folder or in which action of the component the cross-reference object is located.

Target Indicates where the cross-reference object target is located.

Source Indicates where the origin of the cross-reference object target is located.

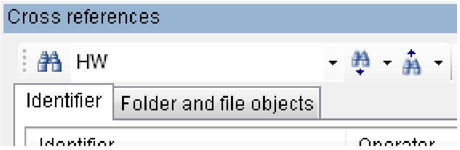
Size Indicates how big the cross reference object is (in bytes).

4.8.3 Search in cross-references

Search options

There are two ways of searching for cross-references: The specified character string is searched for in all columns and rows.

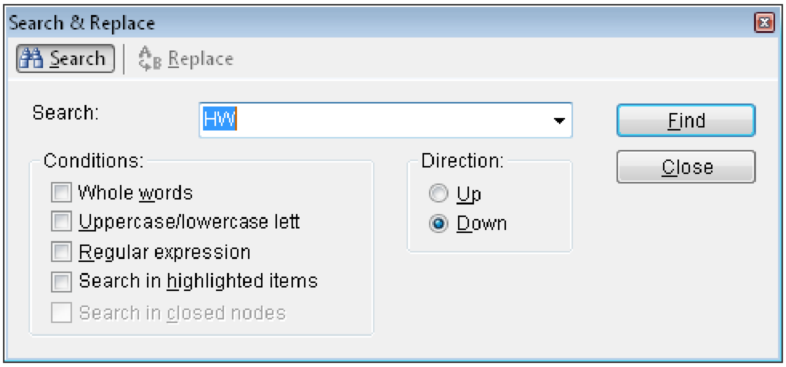
Quick search



HW

By entering in the search field and clicking the "Search up/search down" button or pressing Enter.

Extended search using the search dialog



Search: HW

Conditions:
- □ Whole words
- □ Uppercase/lowercase lett
- □ Regular expression
- □ Search in highlighted items
- □ Search in closed nodes

Direction:
- ○ Up
- ● Down

The extended search dialog is opened by clicking on the binoculars located to the left of the search field.
Under Conditions / Direction, you can define the various search properties.

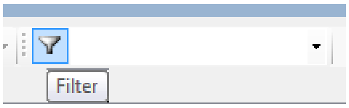
4.8.4 Filter according to identifier or object

A partial string can be entered in the input field by which the column "Identifier" or "Object" is filtered. All objects are listed that contain this character string at any position in the name.

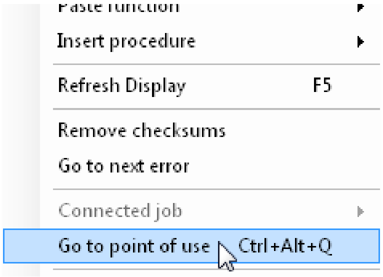
Filtering is reset by entering an empty string.

Procedure

If you want to filter the cross-references according to a certain string, proceed as follows:

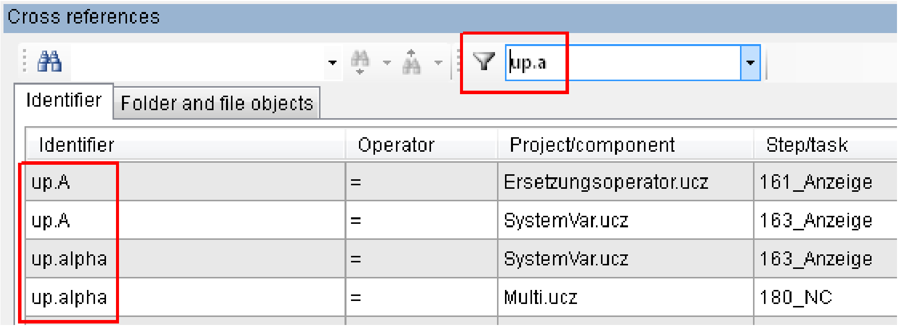


Enter the desired string in the field using the keyboard or select "Go to position of use" from the shortcut menu in the instruction editor to filter a selected identifier.



Filtering is applied to the "identifier" or "object" column.

Example:



Identifier	Operator	Project/component	Step/task
up.A	=	Ersetzungsoperator.ucz	161_Anzeige
up.A	=	SystemVar.ucz	163_Anzeige
up.alpha	=	SystemVar.ucz	163_Anzeige
up.alpha	=	Multi.ucz	180_NC

4.9 Password protection

4.9.1 Overview

Create MyConfig offers password protection for projects, components, packages and logbooks. Using this password protection function, you can protect your own know-how from being viewed and changed by unauthorized personnel.

You can use the password list to define various access levels. You can then execute the following activities only by entering the correct password:

- Opening or saving a project or a component.
- Executing a package at the machine or reimporting this package as project.
- Viewing a protected logbook.

Project structure with access levels

The project and package structure with the possible access levels is shown in the following overview diagram:

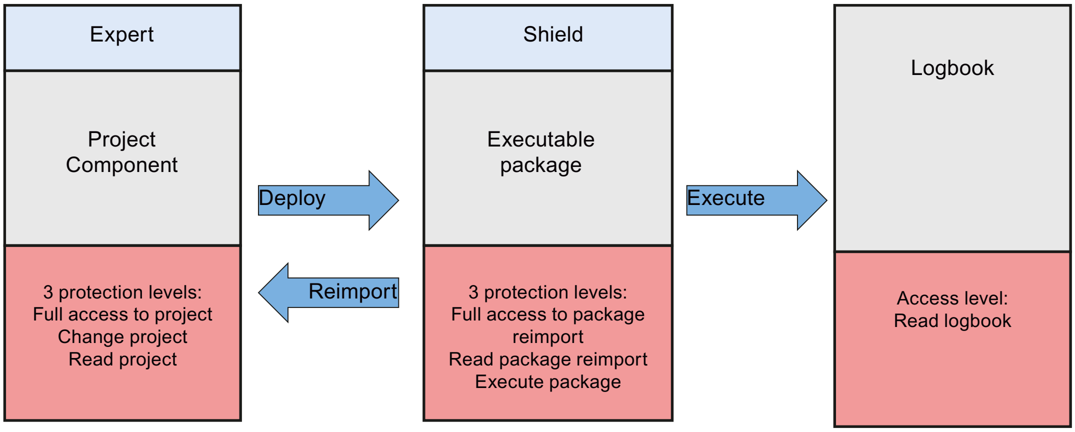


Figure 4-74 Systematically displaying the project structure with the associated access levels

4.9.2 Protecting projects and components

4.9.2.1 Access levels

In Expert, a CMC project is protected using three access levels with different authorization levels. The same access levels are also applied for components. It is necessary to specifically protect components, because components can be opened and processed independently of projects.

You can only view protected CMC projects or protected components if you have a password with one of these access levels.

Converted projects from older versions or newly created projects are unprotected and can be edited as previously. You can subsequently provide projects with protection at any time.

To enter a password that permits defined processing permissions, use "Project > Password protection > Enter password...."

Access levels

Access level	Authorizations
"Read project"	• Opening a project/component for reading
"Change project"	• Opening a project/component for reading • Changing and saving a project/component
"Full access to project"	• Opening a project/component for reading • Changing and saving a project/component • Editing a project/component password list - transfer password list to components

4.9.2.2 Editing a password list

Requirement

You define the password protection already during the creation of a project in the Configuration tab under Project - General > Access.

To make changes at a later time, e.g. activate password protection or manage the password list, use the main menu in Expert: "Project > Password protection > Edit password list...". The dialog "Edit password list [Rights without a password]" is opened.

If there is already password protection, then you will be initially asked to enter the password for "Full access".

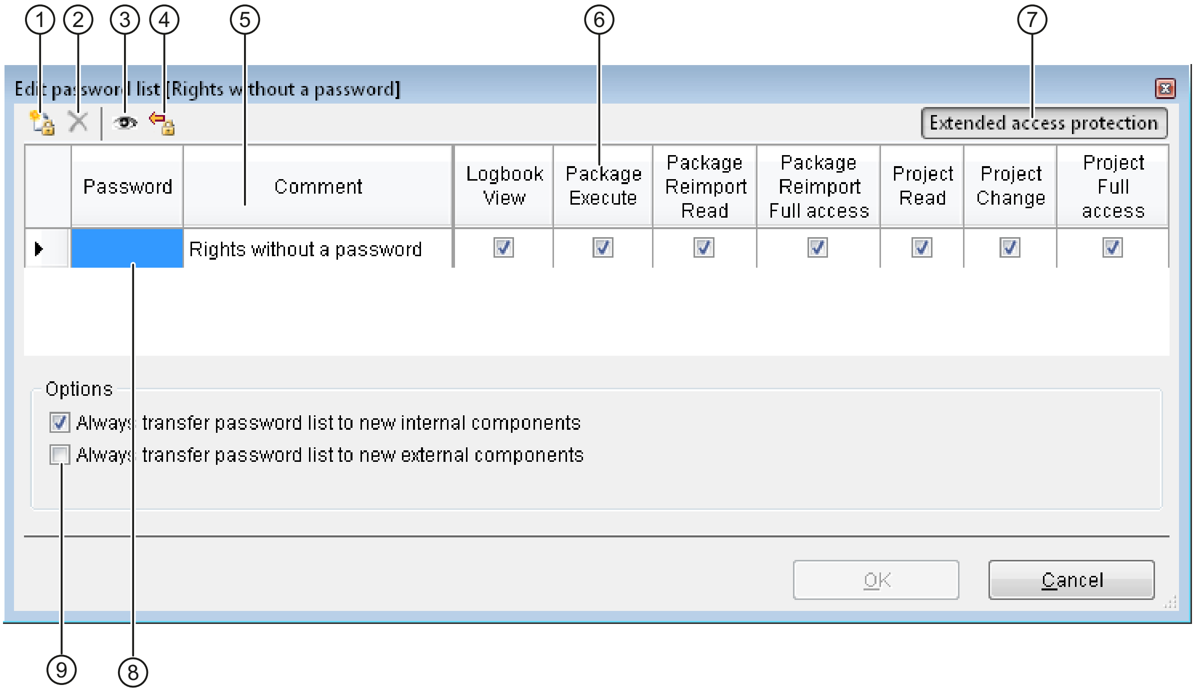
By default, when creating a new project or an external component, an entry <Right without password> will be created with the following authorizations.

Placeholder screenshot - Edit password list [Rights without a password] (by default with 4 authorizations)

You can select the maximum authorizations via the Extended access protection button.

If, when opening the password list, other passwords exist with the access level "Full access," these cannot be viewed or modified, however they can be deleted.

Overview of user interface



Edit password list [Rights without a password]

	Password	Comment	Logbook View	Package Execute	Package Reimport Read	Package Reimport Full access	Project Read	Project Change	Project Full access
▶		Rights without a password	☑	☑	☑	☑	☑	☑	☑

Options

☑ Always transfer password list to new internal components

□ Always transfer password list to new external components

① Add new password"
You can add any number of entries to the list using this button. Passwords must be unique.

The new entries that are added are allocated default passwords: the first entry is 001, the second 002 etc.

These passwords should be changed in order to guarantee the appropriate level of security.

② Remove password"
You can remove the selected entries using this button.

③ Display password"
You can display the password in plain text using this button. All passwords with "full access" are displayed encrypted so that other project engineers cannot see these passwords.

④ Import password list:
You can import the password list of another project using this button.
In the window that opens, you must then search and open the corresponding project.
The current password list is deleted on importing.

⑤ Comment field:
You can add any text to the particular entry in the comment field. Here, we recommend that you describe the role or the person who is using this password with the appropriate access levels. Comments must be unique.

⑥ Selecting access rights:
Using the option boxes, you can activate or deactivate access rights for certain passwords.

The reduced view only offers access levels for the package. The extended view (see ⑦) allows you to allocate access rights for the project.

⑦ Better access protection:
If you wish to protect your project/your component within the company, then you can extend the access rights.

There must always be at least one password with full access. Expert checks that this really is the case. This means that it is not possible to completely lock a project / a component as a result of an incorrectly configured dialog.

⑧ Allocating a password:

Here, you can assign a password with up to 24 characters for specific access permissions.

⑨ Options

- Password list is always transferred to new internal components:
  If a new, internal component is created within the project, the password list of the project for this new component will automatically be transferred.
- Password list is always transferred to new, external components:
  If a new, external component is created in the project, the password list of the project for this new component will automatically be transferred.

See also

Package (Page 92)

4.9.2.3 Transferring a password list to existing components

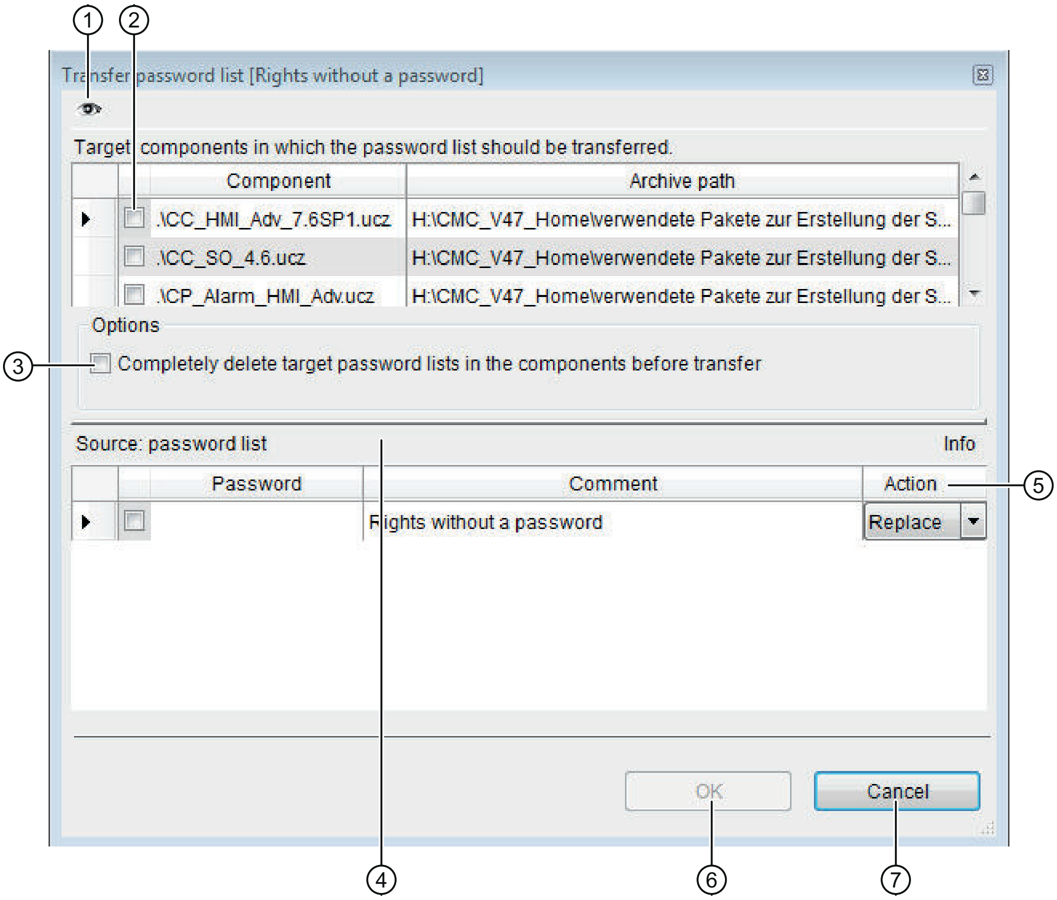
Applications

The relevant dialog for transferring the password list is started via the main menu "Project / Component > Password protection > Transfer the password list > To project components... / to components of an archive folder...".

There are various applications, where it makes sense that you manually transfer your password list to your components, for example, if you have:

- Created a project with a lot of internal linked, internal unlinked as well as external linked components.
- Created a new password list to protect the project or have changed it. You want to transfer this password list to all of the components of your project.
- Created many unlinked components. You want to transfer subsequent changes to the password list to these components.

Overview of the user interface



Transfer password list [Rights without a password]

Target components in which the password list should be transferred.

		Component	Archive path
▶	□	.\CC_HMI_Adv_7.6SP1.ucz	H:\CMC_V47_Home\verwendete Pakete zur Erstellung der S...
	□	.\CC_SO_4.6.ucz	H:\CMC_V47_Home\verwendete Pakete zur Erstellung der S...
	□	.\CP_Alarm_HMI_Adv.ucz	H:\CMC_V47_Home\verwendete Pakete zur Erstellung der S...

Options

□ Completely delete target password lists in the components before transfer

Source: password list

Info

		Password	Comment	Action
▶	□		Rights without a password	Replace

① **Display password**
If this is activated, the passwords are displayed in plain text. Passwords permitting full access are an exception to this.

② Selecting the target components.

③ Option **Delete target password lists in the components completely before transfer**
If this option is selected, the password list will be deleted completely in the target components and then the selected passwords will be transferred with the associated rights.

④ **Password list**
Displays the passwords and the associated comments from the password list of the project or the source component.

Here, you select the passwords that are to be transferred.

The selected passwords are transferred to the target components, depending on the selected action.

⑤ **Action**
The desired actions are defined via the checkboxes:

- Add
  The password is added to the target component with the associated rights.
  If the password already exists, the rights are overwritten additively.
- Replace
  If an identical password is found in the target component, only the rights of the source password are applied.
  If a password to be transferred to the target component does not exist, no action will be performed.
- Delete
  The password is deleted in the target components.
  The first line "Rights without password" cannot be deleted.

⑥ With the **OK** button, transfer of the selected passwords ② to the selected components is performed according to the selected action ④.

⑦ With the **Cancel** button, you discard the settings and close the dialog box.

Note

Also observe the transfer of rights for "Rights without password"

If you have also changed the rights of the "Default password" for the "Rights without password" in your source for the transfer (project or component) and also want to transfer this change, you must also activate this password for the transfer. Generally, "Replace" makes sense as action.

This means that if you create new passwords in your password lists and transfer rights from "Rights without password" to them, you remove these rights from the "Rights without password".
Make sure during the transfer that you not only transfer your new passwords, but also the "Rights without password" as otherwise this would still retain all rights and your components would not be protected.

4.10 Create MyConfig packages

4.10.1 Introduction

How can Create MyConfig packages be processed or executed?

Create MyConfig packages can be created for Windows and Linux systems with Expert.
For the Linux operating system on the NCU, Expert generates a file with the extension "usz" (Linux package) and for the Windows operating system on the PCU, PG or PC, a file (Windows package) with the extension "exe".

The following sections describes the procedure for the Windows and Linux operating systems.

4.10.2 Executing the package under Windows

4.10.2.1 Executing a standard package

Precondition

If the package to be executed under Windows should also access the data of the NCU (the "NCU" area is activated in the package), then the NCU must access a shared folder on the Windows computer (PCU/PG/PC) on which the package is executed.

Note

If an active firewall prevents access from the NCU to the PG/PC, then the package cannot be executed from the PG/PC.

In this case, the package can be executed on the PCU50 or the package must be transferred as a Linux package (see Executing the package under Linux (Page 249)).

The following is required for the execution of (Windows) packages that also access data of the NCU:

- A network connection to the NCU / PCU
- Create MyConfig NcuShareService
  For package execution, prior to establishing the connection to the NCU, it is checked if Create MyConfig NcuShareService is available on the Windows system with a sufficiently up-to-date version.
  If an insufficient version is found, the package execution installs a current version. (This is part of the package).
  During the package execution, the folder C:\ProgramData\Siemens\UpNcuShare is released as an exchange release by Create MyConfig NcuShareService.
  When package execution is finished, the release is canceled by the service.

NcuShareService

The NcuShareService service is required for executing CMC packages with NCU share by Windows PC or by PCU50. The service activates an exchange release (share) on the Windows system for the duration of the package execution. There are three possible ways to install the service NcuShareService on the Windows PC (PG or PCU):

1. Automatically during installation of CMC 4.7
2. Manual installation NcuShareService
3. After enabling of the service by configuration.

In Expert, the service is installed by Shield at the start of a CMC package if it does not already exist and it can be configured and even uninstalled later.

Starting to execute the package under Windows

Establish a network connection between the PC and the control devices (NCU7x0, PCU) that is to be processed with the package.

The package generated by Expert for Windows with the file extension "exe" can be started on the PCU, the PG or the PC using standard Windows resources ("Run...", double-click, etc.).

If the file is the only exe file in the folder "Install" of a USB medium inserted in the PCU (USB FlashDrive, hard disk, CD), then the file is automatically started when the PCU is booted.

Note

Unknown USB medium

For older PCU-base systems, ensure that the USB medium is known to the PCU, 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 PCU again after approximately 30 seconds.

Note

In a PCU with Windows 7, various settings are required to permit package execution for automated commissioning.

Expert provides support with generating a configuration file for initial startup of the PCU (as-delivered state).

In project in which you select the PCU area and Windows 7 operating system, this support will be provided automatically.
The necessary settings are stored in the file unattend.xml. The file must be transferred to a USB medium and plugged in before initial startup of the PCU.

Note

To be considered during initial startup of the PCU50 with Windows 7

If no configuration file unattend.xml was used during the commissioning for the first boot-up of the PCU50 with Windows 7, no update packages will be executed later from USB media from the Install folder.

Packages can only be executed locally from the PCU from the service desktop.

4.10.2.2 Executing the package on the PCU 50 from the network

Windows packages can be executed on the PCU 50. The package is on a network drive.

Preconditions

The following requirements apply:

- There must be a connection to the network drive.
- The folder containing the Windows package must be released for a user of the PCU 50 (default user "auduser") for read and write access.
- If the Windows package contains a restart of the PCU 50 (e.g. due to an installation), then the PCU-local default user "auduser" and the password "SUNRISE" must be used.

Procedure

1. Transfer the project in Expert as Windows package.
2. Save the Windows package in a folder of the network drive.
3. Release the folder of the network drive for users of the PCU 50 (preferred default user "auduser").
4. Let the PCU 50 start up in the Service Desktop.

   Note

   Do **not** start HMI Advanced or SINUMERIK Operate.

5. In the Windows Start menu, select "Start > Run".
6. Enter the following command: "\\<IP address or UNC name>\<share name>\<package name>.
   e.g.: \\10.8.6.16\Package folder\Machine_1.exe
7. Follow the instructions of the package dialogs.

See also

Executing a standard package (Page 247)

4.10.2.3 Package execution dialogs

A representation of the package execution dialogs can be found at Dialogs (Page 96).

4.10.3 Executing the package under Linux

4.10.3.1 Introduction

Expert can deploy the package to a "normal USB FlashDrive" (or hard disk or CD) or a service USB FlashDrive from Siemens. Both versions result in a different system behavior which is subsequently explained.

Package execution from "normal USB FlashDrive":

If the package was transferred to a "normal USB-FlashDrive" (or hard disk or CD) which cannot be booted because it does not contain a Linux boot system from Siemens, then the NCU starts Linux from the CompactFlash card and then starts the package on the USB medium.
For this version, a Linux system that can be run (system software SW) must be available on the CompactFlash card. The system software can be upgraded or reinstalled by the package.

Package execution from the service USB FlashDrive:

If the package has been transferred to a service USB flash drive (from Siemens) that is bootable because it contains a Linux boot system from Siemens, the NCU will start the Linux from this USB flash drive and then start the package on the USB flash drive.
In this variant, the CompactFlash card does not have to contain an executable Linux system (system software SW), i.e. the CompactFlash card can also be empty.

Note

In this case, reinstallation of the system software is a precondition for package execution.

As soon as installation on the CompactFlash card has been completed, the package will restart the NCU and force Linux to start from the compact flash card.

If "None" or "Upgrade" is selected in the "NCU system software" dialog box and there is a system on the CF card, starting is performed from the CF card during execution of this dialog box.

Package execution then continues the same as for a "normal USB FlashDrive".

4.10.3.2 Starting package execution under Linux

The package generated by Expert for Linux with the file extension "usz" should be transferred into the root directory of a USB medium (USB FlashDrive, hard disk) – and this medium should be inserted into a direct USB interface (X125 or X135) of the NCU 7x0. The package is automatically started when the NCU boots.

If there are several packages in the start folder, they will be provided on the Package selection dialog page.

If you select a package other than the default package for execution, clicking the Next button ends the default package and starts the newly selected package.

The result of this procedure is that the VNC server of the NCU is restarted. In this context, there is a loss of connection to all VNC viewers.

The VNC viewer running on the TCU is then automatically ended and restarted.

An external VNC viewer running under Windows on your PC/PG or the PCU might possibly have to be restarted if no automatic connection attempts were set for this.

4.10.3.3 Executing the package on NCU from the CompactFlash card

Linux packages can be executed from the CompactFlash card of the NCU that contains the CNC software under certain circumstances.

Preconditions

The following requirements apply:

- The CompactFlash card must contain a bootable CNC system.
- Fresh installation or upgrading is not permissible.
- The Linux package <name> can be stored under any path <path> on the CompactFlash card. We recommend storage according to the responsibility below "/card/oem" or "/card/user".

Procedure

1. Create an "sci.ini" file with the entry: "Start=<path>/<name>".
   Example: Start=/card/user/Test.usz
2. Save this file in the "/<responsible>/system/etc" folder of the CompactFlash card.
   <Responsible> can be "siemens", "addon", "oem" or "user".
3. Click on main switch "Off/On".
4. Follow the instructions of the package dialogs.

Note

The "sci.ini" file is deleted from the package as soon as it has been started. This will ensure that a package is not unintentionally executed several times.

4.10.3.4 Executing the package on the NCU from the network

Linux packages can also be executed from a network drive.

Procedure

1. Create an "sci.ini" file with the entry: "Start=//<domain>/<user>%<password>@<Server>/<release>/<package name>.usz".
   The specification of the domain is optional.
   Example: Start=//wwprod/user%pass@192.168.100.2/Share/NCU_Pack.usz
2. Save this file in the "/<responsible>/system/etc" folder of the CompactFlash card.
   <Responsible> can be "siemens", "addon", "oem" or "user".
3. Click on main switch "Off/On".
4. Follow the instructions of the package dialogs.

Note

The "sci.ini" file is deleted from the package as soon as it has been started. This will ensure that a package is not unintentionally executed several times.

Note

The most important mechanisms for creating a user and performing a share under Windows are described in Sections Auto-Hotspot and Auto-Hotspot. You can use these or any other releases/users for the package execution from the network.

4.10.3.5 Package execution dialogs

A representation of the package execution dialogs can be found at Dialogs (Page 96).

4.10.4 Reimporting packages

4.10.4.1 Overview

With Expert, packages can be reimported via "Project > Import...", i.e. the project used as basis is regenerated. The complete know-how of the project engineer is in this project. To protect this know-how and prevent unauthorized use of a transferred package, three access levels can be assigned with different authorizations.

Therefore, a project development engineer can make the know-how of his package accessible to a group of previously defined people.

The procedure is described in Section "Password protection (Page 240)".

You can only execute or view (reimport) protected CMC packages if you have a password with one of these access levels.

Access levels

Access level	Authorizations
"Execute package"	• Execute package
"Read package reimport"	• Only open project for reading after a reimport
"Full access to package reimport"	• Open a project for reading • Change or save project • Edit password list of project

4.10.4.2 Reimport with password protection

Preconditions

In principle, it is always possible to reimport a package of the same version. However, to open the project that has been created, you require a password which gives you the permission "Read reimported package" or "Full access to reimported package."

Rules for a reimport

- When opening a reimported project or a reimported component, all passwords are removed that are not identical to the password used for opening.
  The entry <permissions without password> remains unchanged with its configured rights.
  In the result, in the password list, there are a maximum of 2 entries; the entry
  <rights without password> and the entry of the password used.

  Note

  This point is only implemented, if you logged on with a password for "Package reimport full access" and the project was subsequently saved.

- If the password used has the permission "Package reimport full access," in addition to this permission, after the reimport, all of the other permissions to the project/component will also be assigned to it.
- If the password used only has the permission "Package reimport read," no other additional permissions are allocated.
  The project or the component cannot be stored and any passwords with "Full access to reimport" are therefore retained.
- No permissions for reimporting a component have been granted, it can be deployed but not viewed or modified.
  Reimported components that have permissions for the project password used are opened with this permissions without an additional query.
  Otherwise, the password protection behaves in the same way as for reimported components.

4.11 Command line call

4.11.1 Expert

Create MyConfig Expert can be started in the command line mode for the package deployment. This is designed for the automated deployment of finished packages.

A project configuration can be optionally specified during the deployment.

Expert can import a project configuration before the deployment of a project. The project will then be deployed with the imported settings.

Call options

Import project configuration

In order to be able to import a configuration before the deployment, the following minimum syntax must be adhered to in the command line:

```
/pack <Expert project path> /config <Absolute file path of the
configuration file> /deploy /silent
```

The settings imported by the configuration file can be replaced again by arguments of the respective switch, where the current switch comes into effect:

- /deploydir
- /deploy
- /target

Automated forwarding of the project to the package

`Expert <project path>`	
`Expert [/pack] <project path> /deploy [<deployment name>]`	
	`/deploydir <deployment directory>`
	`/silent /target (NCU | PCU | UAZ) [/protect] [/log <Log file>]`
	`[/tgz <tgz file>] [/sdb <Path of the sdb file>] [/ust <Path of the ust file>] [/utz <Path of the utz file>]`
	`[/uss <path of uss file>]`

Note

Umlauts and special characters

Avoid the use of umlauts and special characters in path names.

If special characters or umlauts are to be correctly processed in specified paths, batch files with command line calls must be saved 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.

Parameter

Switch	Arguments	Optional	Description
/pack	<"Project path">	No	• The path **must** be specified with the switch /pack. • The switch specifies the path of the package file – either absolute or relative. • The path must be specified in double inverted commas if spaces are included in it (example: "C:\Path with spaces \package file.upz").
/deploy	<"Deploy name">	Yes	• A name of the package for the deployment can be specified. • The switch causes the opened package to be checked and deployed. • No parameters are required for this switch.
/deploydir	<"Deploy directory">	Yes	• The switch specifies the path of the deploy folder – absolute or relative – in which the package should be deployed. The path is the path without the name of the deployed package. • The path must be specified in double inverted commas if spaces are included in it (example: "C:\path with spaces").

Switch	Arguments	Optional	Description
/log	<"Log file">	Yes	- The switch specifies the path of a log file – either absolute or relative. - Alarms and faults are written to the log file (XML format) when checking and deploying the package. - The path must be specified in double inverted commas if spaces are included in it (example: "C:\Path with spaces \Expert_Logfile.log"). - If the switch is not specified, the log file with the name of the package in the path for deployment is automatically saved.
/silent	-	No	- The graphical user interface of Create MyConfig Expert is hidden.
/config	<"absolute path to the project configuration file">	Yes	- The switch /config specifies the configuration file (with extension) under an absolute path. - The configuration file can have any file extension. Expert then attempts to import the file and apply the project settings. **Note** Project and package configuration files are normally exported/ saved as *.upcfg via Expert.
/target	< NCU | PCU | UAZ >	No	- The switch defines the target system where the package will be deployed. - If UAZ or no value is specified, then the package is automatically deployed without executable part.
/uss	<"Path of the USS file">	Yes	- The switch specifies a USS file that is added and selected during deployment.
/ust	<"Path of the UST file">	Yes	- The switch specifies a ust file that is added and selected for deployment.
/tgz	<"Path of the TGZ file">	Yes	- The switch specifies a tgz file that is added and selected during deployment.
/sdb	<"Path of the SDB file">	Yes	- The switch specifies an SDB file that is added and selected during deployment.
/utz	<"Path of the UTZ file">	Yes	- The switch specifies a UTZ file that is added and selected during deployment.
/protect	-	Yes	- Automated "always reimport protected" deployment of packages.

Return values

Note

The return values for the Expert command line call only apply if Expert is executed in /silent mode without the user interface.

Value	Description
0	No faults or alarms occurred when checking/deploying the package.
1	Alarms, but no faults occurred when checking/deploying the package.
2	Faults (and possibly alarms) occurred when checking/deploying the package.
3	Incorrect command line call.

Example

(Requirement: Create MyConfig is installed in the directory "C:\Program Files\SIEMENS \Create MyConfig 4.7" and a package "Pack_1.upz" is available in the path "C:\Packages\".)

```
Set PATH=%PATH%;"C:\Program Files\SIEMENS\Create MyConfig 4.7"

Expert.exe /silent /pack "C:\Packages\Pack_1.upz" /deploy /deploydir "D:
\Deploying" /target PCU

if %ERRORLEVEL% EQU 3 (
    @echo "Error when calling Expert.exe!"
    pause
)

if %ERRORLEVEL% EQU 2 (
    @echo "Faults during deployment!"
    pause
)
if %ERRORLEVEL% EQU 1 (
    @echo "Alarms during deployment!"
    pause
)
if %ERRORLEVEL% EQU 0 (
    @echo "Deploy executed without error!"
    pause
)
```

4.11.2 Package

Call

```
<package> [-hide] [-batch] [-pw:<pass>]
```

Parameter

Options	Arguments	Description
-hide	-	• This causes the package to be processed without being displayed. • Implies in the "-batch" option.
-batch	-	• This causes the package selection and the completion message to be bypassed for packages with NCU configuration.
-pw	<pass>	• Uses the password "pass," forwarded in plaintext, as input for "Execute package."

Return values

Value	Description
0	No faults have occurred.
1	Cancellation by user
2	Abort through step script or script function.
3	Incorrect command line call.
11xx	Dialog xx could not be advanced in automatic or progress mode because a fault occurred.
12xx	Dialog xx could not be advanced in automatic or progress mode because there is a configuration error on the dialog.
13xx	The dialog page xx could not be manually activated further during package start using the option -hide. There is a configuration error on the dialog page.

Example

```
package.exe -batch
```

4.12 SINAMICS device configuration

4.12.1 Production of modular machines

The Create MyConfig (CMC) engineering software supports both the automated production of modular machines for the machine manufacturer as well as the upgrade of machines for end customers, which requires only a limited number of operator actions.

The production of machines with 840D sl requires a basic commissioning of the SINAMICS, generally designated as "System configuration". The system configuration can be performed with CMC either as "Automatic configuration" or as "User-specified topology."

When the system configuration is configured in the CMC package, all the data sets and optimization data in the step tree required within a series of modular machines in the maximum machine configuration can be specified. 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.

4.12.2 SINAMICS device configuration with Automatic configuration or user-specified topology

Applications

A differentiation is made between three applications that CMC supports when configuring SINAMICS:

1. The SINAMICS was commissioned previously with SINUMERIK Operate or with a different commissioning software. The existing drive data can be checked with a CMC package via "Initial state" (online access) on the dialog page "System configuration" under "DRV..." in the drive archive data, including the topology. In addition, a further configuration or optimization of the SINAMICS can be made.
2. The maximum configuration of the SINAMICS is wired on 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. The 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-defined topology file is a SINAMICS offline topology package (SOT) that contains the topology in the final 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 message frames and the BICO links are also entered automatically in both CMC variants.

Table 4-15

Automatic configuration	User-specified topology
Execution via SINUMERIK Operate or CMC (identical result).	Execution only via CMC.
The 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 the automatic configuration are acquired.	The user-specified topology can also include SINAMICS modules that have not yet been wired and deactivated, exception: NCU, NX, CU320.

Automatic configuration	User-specified topology
The DRV archive data also contains the DOs 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 reports an error if the topology specified in the user-specified topology cannot be implemented.

Configuring in Expert

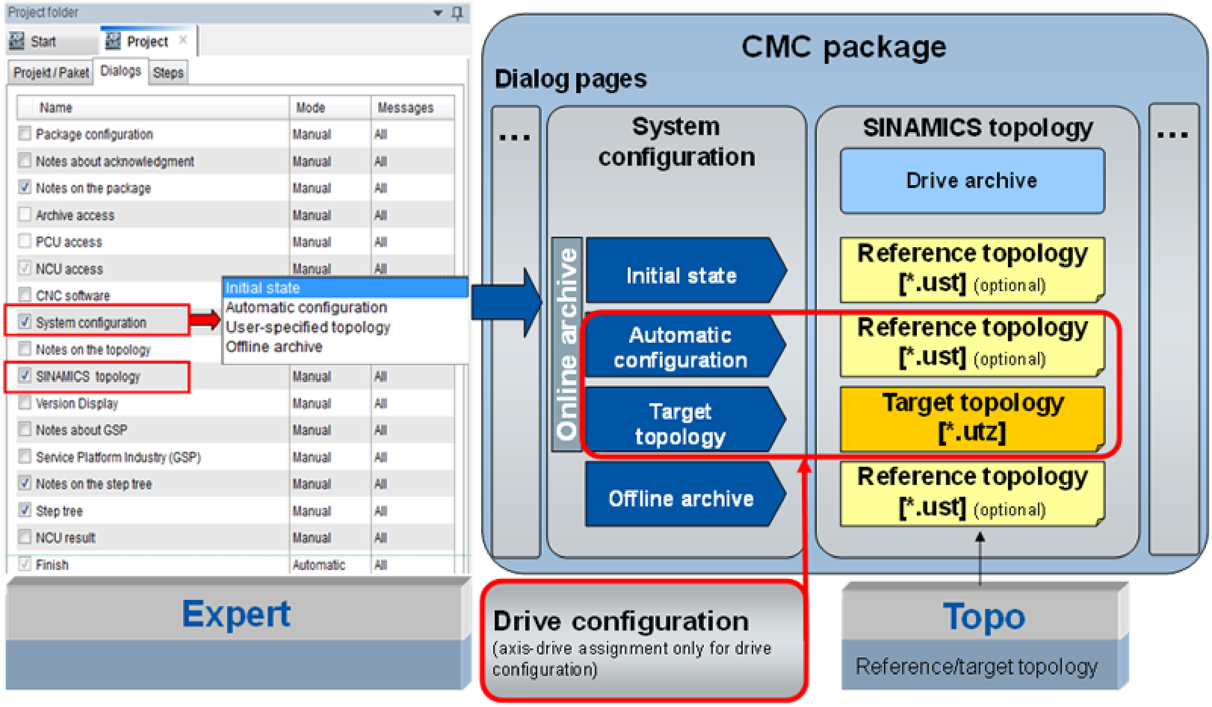
When configuring in Expert, on the "System configuration" dialog page in section "DRV..." under "Origin of data", you can define which of the four applications "Initial state," "Automatic configuration," "User-specified topology," or "Offline archive" is applicable.

By selecting the "User-specified topology" mode, it is necessary to generate with Topo a file with the user-specified topology and 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 "utz."

If "automatic configuration" is performed, as an option, you can use on the "SINAMICS topology" dialog page a file with the comparison topology in XML format and the 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 setting Origin of data" 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 and the comparison topology contain the topology of the SINAMICS and other properties of the DOs and the components.



Project folder

Start | Project

Projekt / Paket | Dialogs | Steps

Name	Mode	Messages
☐ Package configuration	Manual	All
☐ Notes about acknowledgment	Manual	All
☑ Notes on the package	Manual	All
☐ Archive access	Manual	All
☐ PCU access	Manual	All
☑ NCU access	Manual	All
☐ CNC software		
☑ System configuration		
☐ Notes on the topology		
☑ SINAMICS topology	Manual	All
☐ Version Display	Manual	All
☐ Notes about GSP	Manual	All
☐ Service Platform Industry (GSP)	Manual	All
☑ Notes on the step tree	Manual	All
☑ Step tree	Manual	All
☐ NCU result	Manual	All
☑ Finish	Automatic	All

Figure 4-75 CMC package structure

4.12.3 Drive data generation via "automatic configuration"

Basic procedure

The "automatic configuration" via SINUMERIK Operate or via CMC uses the same functions internally. In both cases, the SINAMICS determines its actual topology and saves it as target topology (p9903). This also includes the creation of drive objects (DOs) 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). The file PS000000 is used for the consistence monitoring and the file PS000099 contains configuration data of the internal PROFIBUS. This structure can be saved as drive archive and imported later.

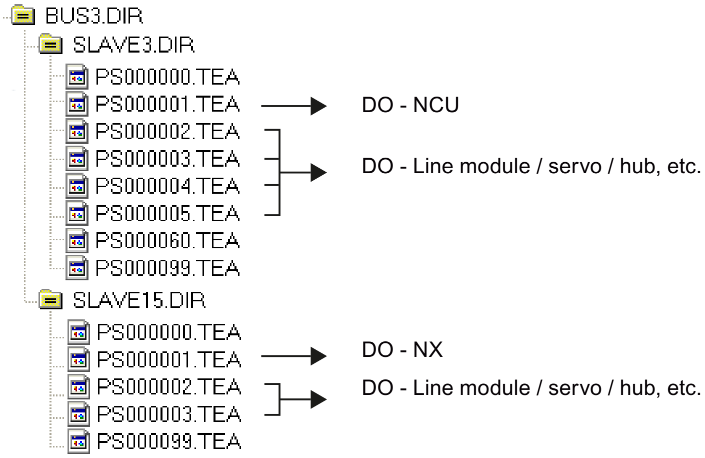


Figure 4-76 Drive data storage

SINAMICS specifies for the "Automatic configuration" DO names, DO numbers, component names, component numbers, etc., and acquires component properties such as article number, component category and component type. The parameters for the DOs are read out of the components, calculated or assigned with factory settings.

The complete data is backed up in the drive data archive.

4.12.4 Drive data generation via "user-specified topology"

Basic procedure

If the SINAMICS device configuration is made with the "User-specified topology," the SINAMICS is assigned a SINAMICS offline topology package corresponding to the maximum configuration of the machine in the form of a user-specified topology file (*.utz) from the CMC package. In other words, components, such as 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 DO names, DO numbers, component names and component numbers as well as component properties such as article number and component type. The structure with generalized drive data is also specified.

Finally, the package processing requests the SINAMICS to check this data based on the currently available configuration and to customize it appropriately, i.e. define the DO or component status. SINAMICS has in the result a customized target topology (p9903) and customized DOs for the specified components.

The DO parameters are customized by SINAMICS with the factory settings, values exported from the components or calculated values. All this data is backed up in the drive data archive.

Because at this time components not yet present in the structure have already been created, the associated user data can be added to the step tree using manipulation tasks. As an alternative, for example with DQI/SMI motors, export of the data from the electronic rating plate with the motor type or encoder type selection can be forced after activation.

4.12.5 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

<table>
<tr><th>Execution step</th><th colspan="2">Result</th></tr>
<tr><td>Export of data from a drive archive in Diff to an ASCII file with reference to the DO in the data storage.</td><td colspan="2">[B3_S3_PS5] (corresponds to the path of the data storage
[BUS3.DIR\SLAVE3.DIR\PS000005.TEA])</td></tr>
<tr><td rowspan="6">Replacement of the reference with a symbolic path.</td><td colspan="2">[$(Up.doX.psPath)]</td></tr>
<tr><td>[…]</td><td>Section for path details in the data storage</td></tr>
<tr><td>$</td><td>Replacement operator</td></tr>
<tr><td>Up</td><td>Identification of a package variable</td></tr>
<tr><td>doX</td><td>DO variable</td></tr>
<tr><td>psPath</td><td>Property that specifies the path in the data storage</td></tr>
<tr><td>Assignment of the DO variables to a DO in the topology.</td><td colspan="2">Manual assignment at runtime on the "SINAMICS topology" dialog page or during the configuration of the topology</td></tr>
<tr><td>Use of the path property after assignment of the doX variables to the DO at runtime.</td><td colspan="2">[$(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)</td></tr>
</table>

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

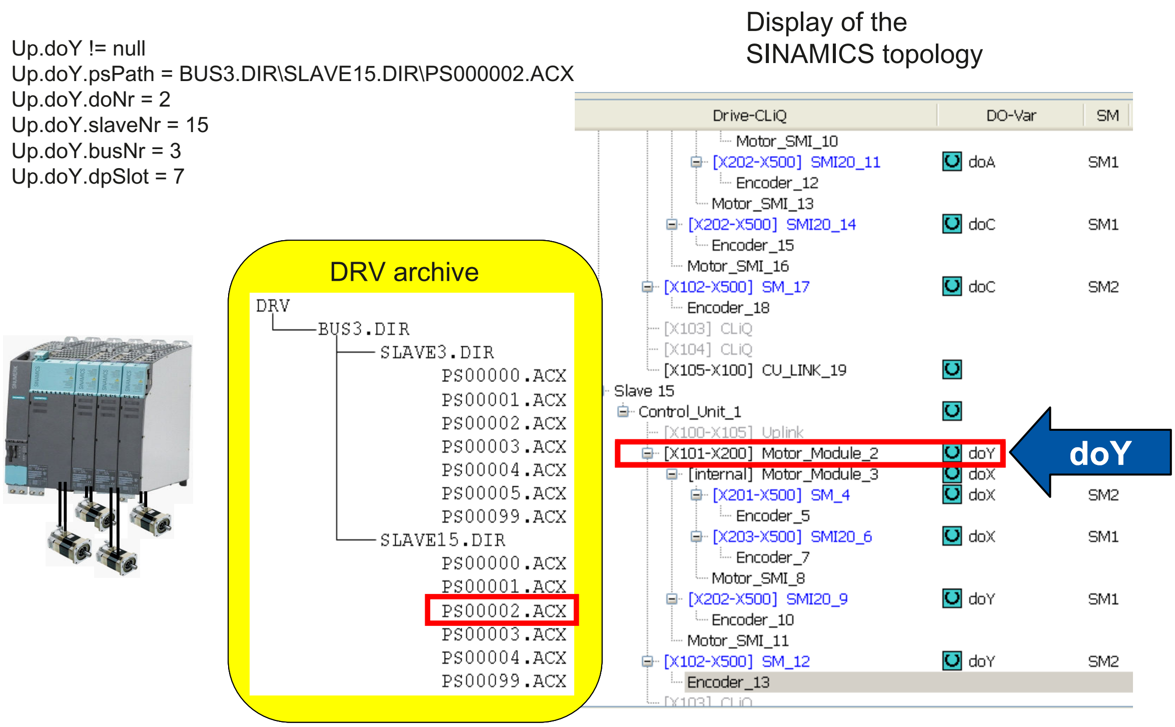
The following properties are available for a DO variable with the name "doVar". The DO variable name "doVar" can be freely selected, for example "doX", "X drive", etc.

Table 4-16 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 manipulation tasks and step scripts.

DO variable with the name doY:



Up.doY != null
Up.doY.psPath = BUS3.DIR\SLAVE15.DIR\PS000002.ACX
Up.doY.doNr = 2
Up.doY.slaveNr = 15
Up.doY.busNr = 3
Up.doY.dpSlot = 7

Figure 4-77 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 two ways of assigning the DO variables to the SINAMICS components:

1. When processing the package, you need to 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.

4.12.6 The advantages of using comparison or user-specified topology

The configuration of a comparison or user-specified topology withTopo 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, context menu, verification 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

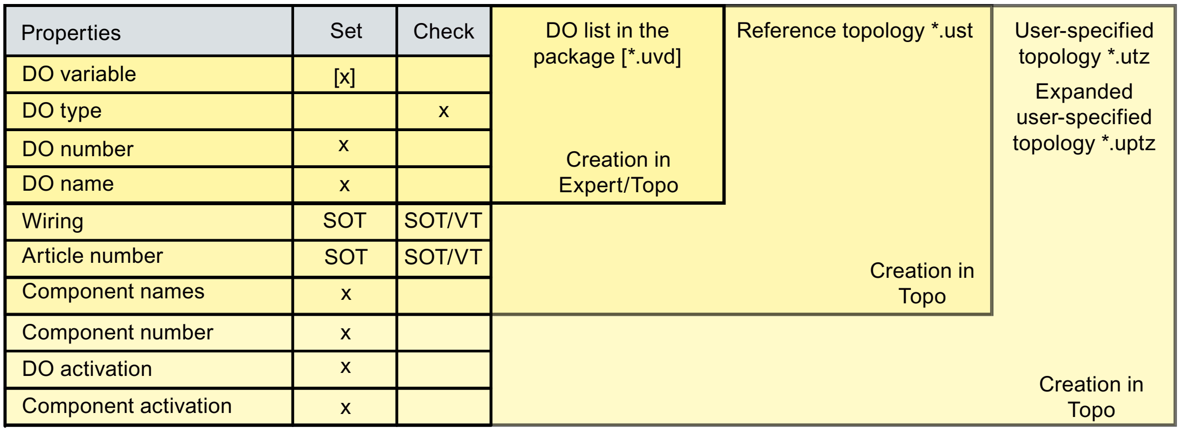
- The actual wiring of the SINAMICS in the control cabinet can be checked using the machine topology exported from SINAMICS. Warning or error messages are issued should deviations occur.
- You can check the properties of the components wired via DRIVE-CLiQ (DO type, article number, etc.). Wildcards are 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. When processing the package, the OEM component number in the DO1 is written to the 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 change of the DO name is configured, then this is automatically entered into the SINAMICS parameter p199 .
  DO number
  With the drive device configuration, SINAMICS assigns the DO numbers for all DOs at the CU and the NXs, i.e. on the particular CU starting at 1. For a machine series, it is more understandable, 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. The value range for the DO numbers lies between 2 and 54.
  Component names
  The component names can be freely assigned. 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. When the user-specified topology is used, the component numbers must also be specified. The CU and the NX must contain 1; the numbers on the CU or NX must not be assigned more than once and must be in the range 2 to 199. In Topo, the component numbers can also be automatically assigned via the shortcut menu.
- When configuring a user-specified topology, the activation status of the DOs and components can still be specified.
  The following is true:
  - 0 inactive and present
  - 1 active and present
  - 2 not present

  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 variable	[x]	
DO type		x
DO number	x	
DO name	x	
Wiring	SOT	SOT/VT
Article number	SOT	SOT/VT
Component names	x	
Component number	x	
DO activation	x	
Component activation	x	

DO list in the package [*.uvd]
Creation in Expert/Topo
Reference topology *.ust
Creation in Topo
User-specified topology *.utz
Expanded user-specified topology *.uptz
Creation in Topo

Figure 4-78 DO list - comparison topology (VT) - user-specified topology (SOT)

4.12.7 Configuration of the topology with Topo

CMC Topo can be used to configure not only a comparison topology but also a user-defined topology. The operation of CMC Topo is described in detail in Section Create MyConfig – Topo (Page 294).

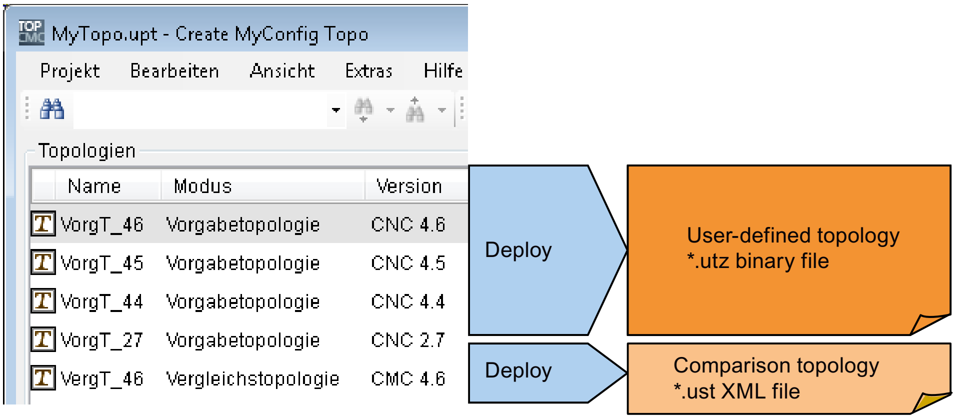
The wiring diagram for the DRIVE-CLiQ connections and some of the properties are identical for both topologies.

With the changeover from comparison topology to user-defined topology, the additional properties DO activation, component activation and component number must also be specified.

On switchover to comparison topology, the data remain stored for when switch-back to user-defined topology is performed. These data have no effect in the comparison topology.

For the deployment as user-defined topology, other necessary data is added internally and output as binary user-defined topology file.

To deploy a user-defined topology, it is necessary to specify a CNC software version.



Name	Modus	Version
VorgT_46	Vorgabetopologie	CNC 4.6
VorgT_45	Vorgabetopologie	CNC 4.5
VorgT_44	Vorgabetopologie	CNC 4.4
VorgT_27	Vorgabetopologie	CNC 2.7
VergT_46	Vergleichstopologie	CMC 4.6

Figure 4-79 Variants of the topology generation with CMC Topo

See also

Generation of the comparison topology using external software (Page 267)

4.12.8 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.

You will find more information on this in Section Interface - Topo-XML (Page 331).

This means that it is also possible to generate the comparison topology using external software, which is integrated in the PLM process of the machine manufacturer, e.g. ePlan, ELCAD or similar. When required, the externally generated comparison topologies can also be read-in in Topo as part of the production preparation for control purposes or adaptations, before they are linked into the project when configuring with Expert or placed next to the package as configuration file. A comparison configuration file, which is located in or next to the package, can also be selected manually on the dialog of the "SINAMICS topology" during the runtime. Normally, a special topology must be created for each machine version, whereas one package can be sufficient for all machines.

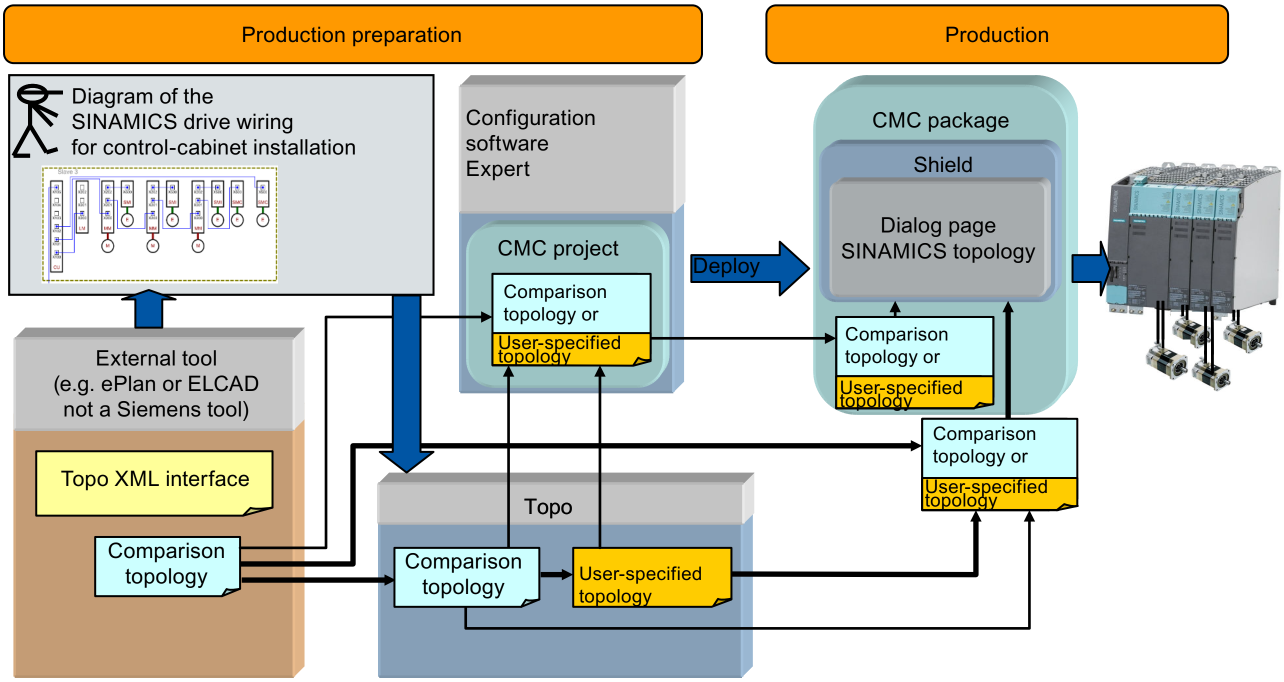


Figure 4-80 Configuration of a CMC package with comparison topology or user-specified topology

Compatibility of CMC versions and CNC software releases

The Topo XML interface is available in the version SCI 2.6, CMC V4.4/4.5 and CMC V4.6. For transfer, CMC Topo V4.6 no longer supports the Topo XML interfaces SCI 2.6 and CMC V4.4/4.5.

Note

CMC Topo 4.6 can only generate comparison or user-specified topology for CMC packages 4.6.

For deployment of a user-defined topology, the CNC system software version must be defined to which this will be applied.
CMC Topo V4.6 supports the CNC software versions V2.6, V2.7, V4.4, V4.5, and V4.6.

4.13 Interface – Step XML

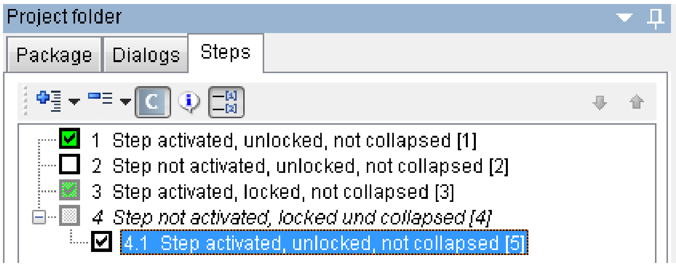
4.13.1 Description Step XML file

Functions of a Step XML file

For an existing package, properties of steps are stored in an XML file (designation: Step XML file or USS file). The assignment of the properties to the particular step is realized using the step ID. When executing the package, a Step XML file in the package or next to the package can be accessed.
The Step XML file can either be output via the shortcut menu in the Expert software or according to the interface description via an external system.
The file must have the extension "<name>.XML" or "<name>.uss".

Property	XML attribute name	Description
Step selection (step activation/deactivation)	checked	• Defines whether the step is to be activated/deactivated. The activation does not mean that it is shown in green and actually executed (see the example below: "4.1 Step is activated and not locked [5]"). Further, it is of no significance as to whether a possibly configured step condition is fulfilled or not fulfilled • Possible values: "True", "False"
Step selection can be changed	locked	• Defines whether the step selection for the operator of the package is blocked or enabled • Possible values: "True", "False"
Expanded/collapsed status	collapsed	• For the time of package execution, it can be defined for each step whether the step will be collapsed. Substeps can be hidden in this way. The overview for display of the step tree can therefore be configured individually. • Possible values: "True," "False"

Step tree configuration file



☑ 1 Step activated, unlocked, not collapsed [1]
☐ 2 Step not activated, unlocked, not collapsed [2]
☑ 3 Step activated, locked, not collapsed [3]
☐ *4 Step not activated, locked und collapsed [4]*
☑ 4.1 Step activated, unlocked, not collapsed [5]

Figure 4-81 Step tree example

```
<upstep packname="MyMachine" version="4.6.0.0">
<step id="1" checked="True" locked="False" collapsed="False" />
<step id="2" checked="False" locked="False" collapsed="False" />
<step id="3" checked="True" locked="True" collapsed="False" />
<step id="4" checked="False" locked="True" collapsed="True" />
<step id="5" checked="True" locked="False" collapsed="False" />
</Upstep>
```

Explanation

`<element>`	Upstep, step	Tag element
`attr=".."`	id="x"	Attribute required for a unique assignment to the step

`attr=".."`	checked="True", checked="False"
	locked="True", locked="False"
	collapsed="True", collapsed="False"
		Optional attributes for a step property

`attr=".."`	packname="Parameter test" version="4.6.0.0"	Optional attributes for assignment to the package, version

Note

The configuration file must not contain properties for each step of the package.
When applying the configuration file to the step tree of the package, only ID-identifiable step properties are accepted.
Additional steps are ignored.
For missing steps, the properties from the step tree in the package are used.

4.13.2 Step xml (USS file)

The "namespace" attribute is designated in the Step xml.

...

4.14 Reference list, system variables

4.14.1 System variables - overview

System variables for Expert

To provide a simple overview, the system variables that you require for the Project folder are listed according to the Package (Page 92), Dialogs (Page 96) and Steps (Page 166) tabs. In addition there is a list for the Environment variables that cannot be changed.

Entry of the system variable is supported by the shortcut menu or by automatic completion function (IntelliSense).

The values of most system variables are provided with values for the first time during configuration. These can be overwritten by using project configuration files during the deployment. Some variables are only available during the package execution.

This is specified for the relevant system variables in the "Description" column.

During the package execution, all of the system variables that are relevant for the package can be overwritten by using a package configuration file. During the package execution as well, system variables can be changed in dialog scripts, step scripts or manipulation tasks by write accesses before the system variables come into effect in the sequence.

For all of the system variables, the time until which a change of the system variables makes sense and is still effective for the package execution is then specified in the respective "Change effective until" column.

The system variables with the entry "ReadOnly" cannot be changed in the script.

4.14.2 Package tab

4.14.2.1 System variables for the Package tab

Package tab

The following system variables are assigned to the Package (Page 92) tab.

- Package properties **Up.$Pack**

The enumerations are marked with the icon and the elements of the enumerations with the icon .

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.

4.14.2.2 Enumeration runtime system - DeployTargets

Range	System variable / enumeration	Type	Change effective until	Description
	Enumeration runtime system - DeployTargets			
	Up.$Pack.DeployTargets.ADDON	Enumerator value	readOnly	Package without the executable part
	Up.$Pack.DeployTargets.LINUX	Enumerator value	readOnly	LINUX package (NCU)
	Up.$Pack.DeployTargets.WINDOWS	Enumerator value	readOnly	Windows package (PC/PCU)

4.14.2.3 Enumeration PCU operating system - PcuSystem

Range	System variables/enumeration	Type	Change effective until	Description
	Enumeration PCU operating system - PcuSystem			
	Up.$Pack.PcuSystems.WINXP	Enumerator value	readOnly	Windows XP operating system
	Up.$Pack.PcuSystems.WIN7	Enumerator value	readOnly	Windows 7 operating system

4.14.2.4 System variables for package properties $Pack

System variables	Type	Change effective until	Description	Example
Up.$Pack.DeployDir	STRING	readOnly	Storage location	-
Up.$Pack.DeployExclude	BOOL	readOnly	Exclude data storage areas for the project deployment.	false / true
Up.$Pack.DeployName	STRING	readOnly	Name of the executed package. The name is different to the deployment name if the package has been renamed.	-
Up.$Pack.DeployTarget	Enumeration DeployTargets (Page 272)	readOnly	Runtime system	-
Up.$Pack.Language	STRING	readOnly	Language setting	"de"/"en"
Up.$Pack.ARC	BOOL	Package configuration dialog page	Configuration archive selected.	false / true
Up.$Pack.NCU	BOOL	Package configuration dialog page	Configuration NCU selected.	false / true
Up.$Pack.PCU	BOOL	Package configuration dialog page	Configuration PCU selected.	false / true
Up.$Pack.PcuSystem	Enumeration PcuSystems (Page 272)	Package configuration dialog page	PcuSystems	-
Up.$Pack.ProdVersion	VERSION	readOnly	Display of the CMC system version.	"4.7.0.1"

System variables	Type	Change effective until	Description	Example
Up.$Pack.ServiceInstall	BOOL	NCU access dialog page	NcuShareService is installed.	false / true
Up.$Pack.ServiceDeinstall	BOOL	End dialog page	NcuShareService is uninstalled.	false / true
Up.$Pack.UserVersion	STRING / VERSION	readOnly	User version. The type can be a string or version; this depends on the specified value. The default for a blank string is "STRING": `Up.$Pack.UserVersion = ""`	"V1.0" or 1.0.0.2

4.14.3 Dialogs tab

4.14.3.1 System variables for the Dialogs tab

Dialog properties $Dialog

The subsequent system variables are assigned to the Dialogs tab (Page 96) within the project folder. To facilitate a better overview, the system variables are chronologically listed for the particular dialog pages and according to the structure of the context menu.

The enumerations are identified with the symbol ▣, the elements of the enumerations with ▤.

4.14.3.2 Enumeration processing modes - ProcessModes

Range	System variable/enumeration	Type	Change effective	Description
▣	**Enumeration processing modes - ProcessModes**			
	▤ Up.$Dialog.ProcessModes.AUTOMATIC	Enumerator value	readOnly	Automatic editing mode
	▤ Up.$Dialog.ProcessModes.MANUAL	Enumerator value	readOnly	Manual editing mode
	▤ Up.$Dialog.ProcessModes.PROGRESS	Enumerator value	readOnly	Editing mode progress

4.14.3.3 Enumeration message level - InteractLevels

Range	System variable/enumeration	Type	Change effective	Description
Enumeration message level - InteractLevels				
	Up.$Dialog.InteractLevels.ALL	Enumerator value	readOnly	Message level, display all messages
	Up.$Dialog.InteractLevels.ERROR	Enumerator value	readOnly	Message level, only error messages
	Up.$Dialog.InteractLevels.WARNING	Enumerator value	readOnly	Message level, error and warning messages

4.14.3.4 System variables for the Package configuration dialog

Table 4-17 Package configuration dialog - PackageConfig

System variables	Type	Change effective until	Description	Example
Up.$Dialog.PackageConfig.				
Activated	BOOL	readOnly	Activating the dialog processing	false / true
ConfigFile	STRING	Current dialog → OnInit	Configuration preselection	-
InteractLevel	Enumeration InteractLevels (Page 274)	Current dialog → OnInit	Enumeration message level	-
ProcessMode	Enumeration ProcessModes (Page 273)	Current dialog → OnInit	Enumeration processing mode	-

4.14.3.5 System variables for the Notes about acknowledgement dialog

Table 4-18 **Notes about acknowledgement - ConfirmNotes dialog**

System variables	Type	Change effective until	Description	Example
Up.$Dialog.ConfirmNotes.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	false / true
InteractLevel	Enumeration InteractLevels (Page 274)	Last script before dialog	Enumeration message level	-
ProcessMode	Enumeration ProcessModes (Page 273)	Last script before dialog	Enumeration processing mode	-

4.14.3.6 System variables for the Notes on the package dialog

Table 4-19 Dialog, Notes on the package - PackageNotes

System variables	Type	Change effective until	Description	Example
Up.$Dialog.PackageNotes.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	false / true

System variables	Type	Change effective until	Description	Example
InteractLevel	Enumeration InteractLevels (Page 274)	Last script before dialog	Enumeration message level	-
ProcessMode	Enumeration ProcessModes (Page 273)	Last script before dialog	Enumeration processing mode	-

4.14.3.7 System variables for the Archive access dialog

Table 4-20 Archive access - ArcAccess dialog

System variables	Type	Change effective until	Description	Example
Up.$Dialog.ArcAccess.				
Activated	BOOL	Script of the previous active dialog	Activating the dialog processing	false / true
InteractLevel	Enumeration InteractLevels (Page 274)	Current dialog → OnInit	Enumeration message level	-
Location	STRING	Current dialog → OnInit	Archive path - Specify path	-
ProcessMode	Enumeration ProcessModes (Page 273)	Current dialog → OnInit	Enumeration processing mode	-

4.14.3.8 System variables for the NCU access dialog

Table 4-21 Dialog, NCU access - NcuAccess

System variables	Type	Change effective until	Description	Example
Up.$Dialog.NcuAccess.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	false / true
Address	STRING	Current dialog → OnInit	Address or IP address of NCU	`"NCU720"` `"194.168.2.1"`
Archives	BOOL		NCPLCDP archive "Backup.arc" before package run	false / true
Backup	BOOL		Complete data backup (TGZ) before package run	false / true
InteractLevel	Enumeration InteractLevels (Page 274)		Enumeration message level	-
Password	STRING	Current dialog → OnNext	NCU login information: Password	`"SUNRISE"`
ProcessMode	Enumeration ProcessModes (Page 273)	Current dialog → OnInit	Enumeration processing mode	-
UserName	STRING	Current dialog → OnNext	NCU login information: User Name	`"manufact"`

4.14.3.9 System variables for the PCU access dialog

Table 4-22 PCU access dialog - PcuAccess

System variables	Type	Change effective until	Description	Example
Up.$Dialog.PcuAccess.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	false / true
Address	STRING	Current dialog → OnInit	UNC name / IP of the PCU or archive path	`"192,167,140,210"`
InteractLevel	Enumeration InteractLevels (Page 274)		Enumeration message level	-
Password	STRING		PCU login information: Password	-
ProcessMode	Enumeration ProcessModes (Page 273)		Enumeration processing mode	-
UserName	STRING		PCU login information: User name	-

4.14.3.10 Enumeration, installation modes - Modes

Range		System variable/enumeration	Type	Change effective	Description
CNC software dialog - NcuSetup					
	Enumeration installation modes - Modes				
		Up.$Dialog.NcuSetup.Modes.INSTALL	Enumerator value	readOnly	Installations mode, new installation
		Up.$Dialog.NcuSetup.Modes.NONE	Enumerator value	readOnly	Installation mode without installation
		Up.$Dialog.NcuSetup.Modes.UPDATE	Enumerator value	readOnly	Installation mode upgrade

4.14.3.11 System variables for the CNC software dialog

Table 4-23 CNC software dialog - NcuSetup

System variables	Type	Change effective until	Description	Example
Up.$Dialog.NcuSetup.				
Activated	BOOL	Script of the previous active dialog	Activating the dialog processing	false / true
InteractLevel	Enumeration InteractLevels (Page 274)		Enumeration message level	-
Mode	Enumeration modes (Page 276)		Enumeration installation mode	-
ProcessMode	Enumeration ProcessModes (Page 273)		Enumeration processing mode	-
TgzFile	STRING		CNC system software (*.tgz) - preselection	`"./SW/04.05.02.00.007.tgz"`

4.14.3.12 NCSources - enumeration, source of the NC data

Range	System variable/enumeration	Type	Change effective	Description
System configuration dialog - SystemConfig				
	NcSources - enumeration source of the NC data			
	Up.$Dialog.SystemConfig.NcSources.ARCHIVE	Enumerator value	readOnly	Data are taken from an archive.
	Up.$Dialog.SystemConfig.NcSources.FACTORY	Enumerator value	readOnly	Memory reset state of the NC.
	Up.$Dialog.SystemConfig.NcSources.ORIGIN	Enumerator value	readOnly	Actual state of the NC (initial state).

4.14.3.13 PlcSources - enumeration, source of the PLC data

Range	System variable/enumeration	Type	Change effective	Description
System configuration dialog - SystemConfig				
	PlcSources - enumeration source of the PLC data			
	Up.$Dialog.SystemConfig.PlcSources.ARCHIVE	Enumerator value	readOnly	Data are taken from an archive.
	Up.$Dialog.SystemConfig.PlcSources.FACTORY	Enumerator value	readOnly	Memory reset state of the PLC.
	Up.$Dialog.SystemConfig.PlcSources.ORIGIN	Enumerator value	readOnly	Actual state of the PLC (initial state).

4.14.3.14 SdbSources - enumeration source of the SDB data

Range	System variable/enumeration	Type	Change effective	Description
Dialog, system configuration - SystemConfig				
	SdbSources - enumeration source of the SDB data			
	Up.$Dialog.SystemConfig.SdbSources.ARCHIVE	Enumerator value	readOnly	Data are taken from an archive.
	Up.$Dialog.SystemConfig.SdbSources.FACTORY	Enumerator value	readOnly	Memory reset state of the SDB data.
	Up.$Dialog.SystemConfig.SdbSources.ORIGIN	Enumerator value	readOnly	Actual state of the SDB data.

4.14.3.15 DrvSource - enumeration, source of the DRV data

Range	System variable/enumeration	Type	Change effective	Description
System configuration dialog - SystemConfig				
	DrvSources - enumeration source of the DRV data			

Range		System variable/enumeration	Type	Change effective	Description
		Up.$Dialog.SystemConfig.DrvSources.ARCHIVE	Enumerator value	readOnly	Data are taken from an archive.
		Up.$Dialog.SystemConfig.DrvSources.AUTOMATIC	Enumerator value	readOnly	Automatic configuration is used.
		Up.$Dialog.SystemConfig.DrvSources.ORIGIN	Enumerator value	readOnly	The actual topology is used
		Up.$Dialog.SystemConfig.DrvSources.TARGET	Enumerator value	readOnly	A user-specified topology is used

4.14.3.16 System variables for the System configuration dialog

Table 4-24 **System configuration dialog - SystemConfig**

System variables	Type	Change effective until	Description	Example
Up.$Dialog.SystemConfig.				
Activated	BOOL	Script of the previous active dialog	Activating the dialog processing	false / true
DrvFile	STRING	Current dialog → OnInit	Path of the DRV data	`"DRV-Daten.arc"`
DrvProcess	BOOL	Current dialog → OnInit	Use data	false / true
DrvSource	Enumeration DrvSources (Page 277)	Current dialog → OnInit	Enumeration source of the DRV data	-
InteractLevel	Enumeration InteractLevels (Page 274)	Current dialog → OnInit	Enumeration message level	-
NcFile	STRING	Current dialog → OnInit	Path of the NC data.	`"NC-Daten.arc"`
NcProcess	BOOL	Current dialog → OnInit	Use data	-
NCSource	Enumeration NcSources (Page 277)	Current dialog → OnInit	Enumeration source of the NC data	-
PlcFile	STRING	Current dialog → OnInit	Path of the PLC data.	`"PLC-Daten.arc"`
PlcProcess	BOOL	Current dialog → OnInit	Use data	false / true
PlcSource	Enumeration PlcSource (Page 277)	Current dialog → OnInit	Enumeration source of the PLC data	-
ProcessMode	Enumeration ProcessModes (Page 273)	Current dialog → OnInit	Enumeration processing mode	-
SdbFile	STRING	Current dialog → OnInit	Path of the SDB data.	`"./hw/SDB-Daten.arc"`
SdbSource	Enumeration Sdb_Sources (Page 277)	Current dialog → OnInit	Enumeration source of the SDB data.	-

4.14.3.17 System variables for the Notes on the topology dialog

Table 4-25 Notes on the topology dialog - DriveTopologyNotes

System variables	Type	Change effective until	Description	Example
Up.$Dialog.DriveTopologyNotes.				
Activated	BOOL	Script of the previous active dialog	Activating the dialog processing	false / true

System variables	Type	Change effective until	Description	Example
InteractLevel	Enumeration InteractLevels (Page 274)	Script of the previous active dialog	Enumeration message level	-
ProcessMode	Enumeration ProcessModes (Page 273)	Script of the previous active dialog	Enumeration processing mode	-

4.14.3.18 System variables for the SINAMICS topology dialog

Table 4-26 SINAMICS topology dialog - DriveTopology

System variables	Type	Change effective until	Description	Example
Up.$Dialog.DriveTopology.				
Activated	BOOL	Script of the previous active dialog	Activating the dialog processing	false / true
AxisDriveAssignment	BOOL	Current dialog → OnInit	Axis-drive assignment - selection	false / true
InteractLevel	Enumeration InteractLevels (Page 274)	Current dialog → OnInit	Enumeration message level	-
ProcessMode	Enumeration ProcessModes (Page 273)	Current dialog → OnInit	Enumeration processing mode	-
UstFile	STRING	Current dialog → OnInit	Name of the Ust file	`"C:\Data \ust_file.ust"`
UtzFile	STRING	Current dialog → OnInit	Name of the Utz file	`"Utz_File.utz"`

4.14.3.19 System variables for the Version view dialog

Table 4-27 Version Display dialog - VersionView

System variables	Type	Change effective until	Description	Example
Up.$Dialog.VersionView.				
Activated	BOOL	Script of the previous active dialog	Activating the dialog processing	false / true
ProcessMode	Enumeration ProcessModes (Page 273)	Script of the previous active dialog	Enumeration processing mode	-
InteractLevel	Enumeration InteractLevels (Page 274)	Script of the previous active dialog	Enumeration message level	-

4.14.3.20 System variables for the Notes on the Service Platform Industry (GSP) dialog

Table 4-28 Notes on the Service Platform Industry (GSP) dialog - ServiceDataNotes

System variables	Type	Change effective until	Description	Example
Up.$Dialog.ServiceDataNotes.				
Activated	BOOL	Script of the previous active dialog	Activating the dialog processing	false / true
InteractLevel	Enumeration InteractLevels (Page 274)		Enumeration message level	-
ProcessMode	Enumeration ProcessModes (Page 273)		Enumeration processing mode	-

4.14.3.21 System variables for the Service Platform Industry (GSP) dialog

Table 4-29 Service Platform Industry (GSP) dialog - ServiceData

System variables	Type	Change effective until	Description	Example
Up.$Dialog.ServiceData.				
Activated	BOOL	Script of the previous active dialog	Activating the dialog processing	false / true
ProcessMode	Enumeration ProcessModes (Page 273)		Enumeration processing mode	-
InteractLevel	Enumeration InteractLevels (Page 274)		Enumeration message level	-

4.14.3.22 System variables for the Notes on the step tree dialog

Table 4-30 Notes on the step tree dialog - StepSelectionNotes

System variables	Type	Change effective until	Description	Example
Up.$Dialog.StepSelectionNotes.				
Activated	BOOL	Last script before dialog.	Activating the dialog processing	false / true
ProcessMode	Enumeration ProcessModes (Page 273)		Enumeration processing mode	-
InteractLevel	Enumeration InteractLevels (Page 274)		Enumeration message level	-

4.14.3.23 System variables for the Step tree dialog

Table 4-31 Step tree dialog - StepSelection

System variables	Type	Change effective until	Description	Example
Up.$Dialog.StepSelection.				
Activated	BOOL	Script of the previous active dialog	Activating the dialog processing	false / true

System variables	Type	Change effective until	Description	Example
ArchiveBeg	BOOL	Current dialog → OnInit	NCPLCDP archive "StepBeg.arc" before step tree	false / true
ArchiveEnd	BOOL	Current dialog → OnEnd	NCPLCDP archive "StepEnd.arc" after step tree	false / true
InteractLevel	Enumeration InteractLevels (Page 274)	Current dialog → OnInit	Enumeration message level	-
ProcessMode	Enumeration ProcessModes (Page 273)		Enumeration processing mode	-
CfgFile	STRING		Name of the Cfg file for step properties	`"Properties.cfg"`

4.14.3.24 System variables for the NCU result dialog

Table 4-32 NCU result dialog - NcuResult

System variables	Type	Change effective until	Description	Example
Up.$Dialog.NcuResult.				
Activated	BOOL		Activating the dialog processing	false / true
Archives	BOOL		NCPLCDP archive "Result.arc" after package run	
Backup	BOOL		Complete data backup (TGZ) after package run.	
InteractLevel	Enumeration InteractLevels (Page 274)		Enumeration message level	-
ProcessMode	Enumeration ProcessModes (Page 273)		Enumeration processing mode	-

4.14.3.25 System variables for the Finish dialog

Table 4-33 Finish dialog - PackageEnd

System variables	Type	Change effective until	Description	Example
Up.$Dialog.PackageEnd.				
Activated	BOOL	Last script before dialog	Activating the dialog processing	false / true
InteractLevel	Enumeration InteractLevels (Page 274)		Enumeration message level	-
LogDir	STRING		Path to storage location of the logbook.	`Up.$Dialog.PackageEnd.LogDir = "D:\Install"`
LogName	STRING		File name of the logbook	`Up.$Dialog.PackageEnd.LogName = "CMC_Package_Log"`
ProcessMode	Enumeration ProcessModes (Page 273)		Enumeration processing mode	-

4.14.4 Steps tab

Up.$Step[?]

The subsequent system variables are assigned to the Steps (Page 166) tab.

Table 4-34 Steps tab

System variable	Type	Query as of	Change effective until	Description	Example
Up.$Step[?]					

System variable	Type	Query as of	Change effective until	Description	Example
Up.$Step[?].	OBJECT	**Dialog** Package configuration **Event** OnInit	readOnly	Step query whether the step with the specified ID exists. If the specified step ID is not available, "0" is returned. The properties .Activated or .Processing must be used for determining the activation or processing status. **Note** For compatibility reasons, a query for "true" or "false" returns the same result as the new notation Up.$Step[].Processing. The use of the old notation Up.$Step[] of the BOOL type is marked as deprecated in the verification run.	`If Up.$Step[1337] != null`
Activated	BOOL		Last script before current step.	The activation status of the specified step is returned. "true" is signaled when the step is activated, i.e. the checkbox or the option button is set. **Note** This information is returned irrespective of whether the step has a green background (green track) or not.	`If Up.$Step[1337].Activated == true`
Collapsed	BOOL			"true" is signaled when the substeps below the queried step have been collapsed, i.e. are not visible for the operator.	`Up.$Step[1337].Collapsed = true`
Locked	BOOL			Status activation disabled.	`If Up.$Step[1337].Locked == true`
Processing	BOOL		readOnly	The execution status of the specified step is returned. "true" is signaled when the step is activated and has a green background. **Note** Only the green steps (green track) are executed.	`If Up.$Step[1337].Processing == true`

Query step - Up.$Step[id]

The Up.$Step[id] system variable can be used to query whether a specific step is available in manipulation tasks, dialog scripts or step scripts.

Example

```
CHANDATA(1)
If Up.$Step[C_AXIS] != zero      ; Step with ID: C_AXIS is available
  If Up.                         ; Step with ID: C_AXIS is activated
$Step[C_AXIS].Activated == true
    $MA_IS_ROT_AX[AX5] = 1
  EndIf
EndIf
```

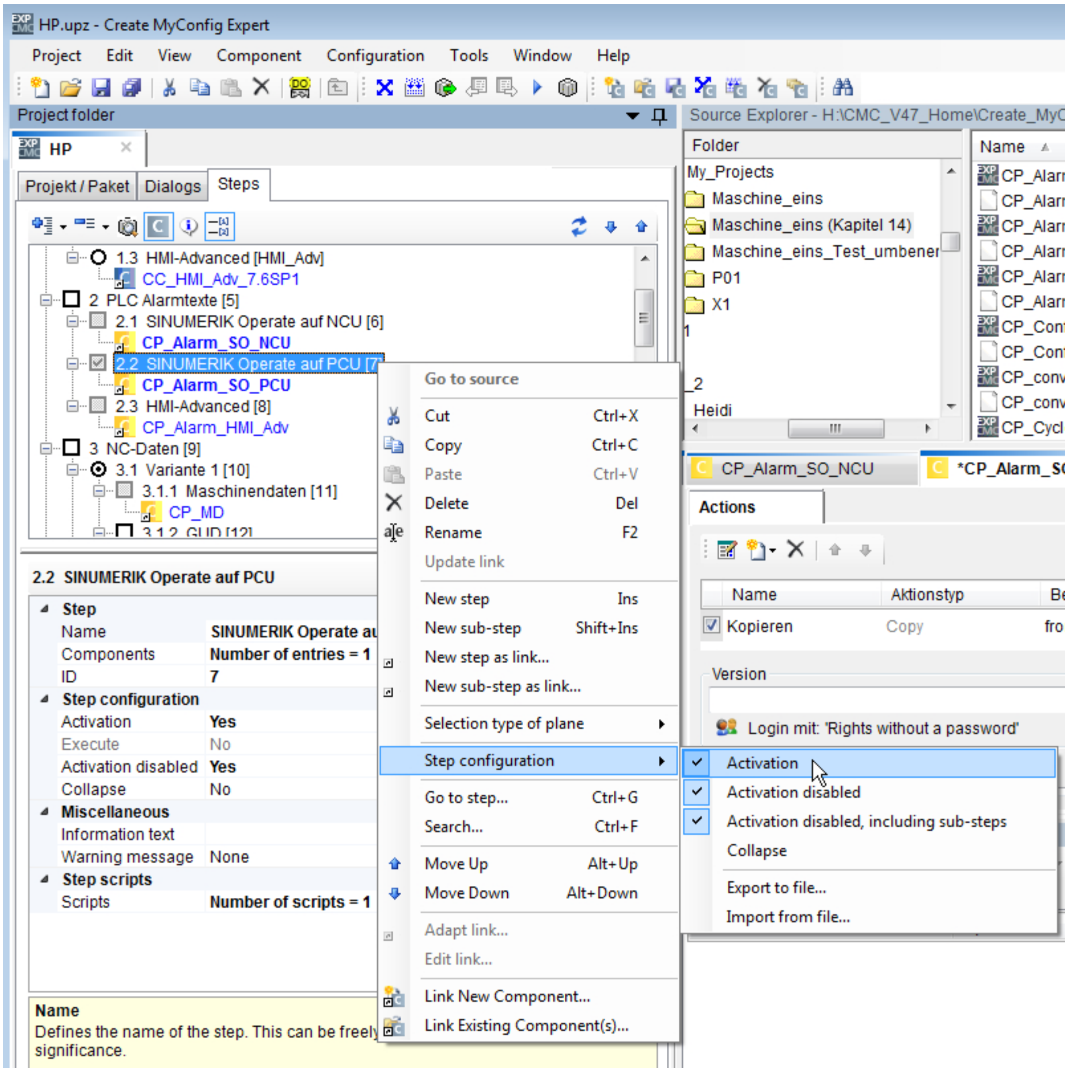
Activate step - Up.$Step[id].Activated

Using the system variable Up.$Step[id].Activated, the activation status of a specific step is defined in the step tree.

The system variable **Up.$Step[id].Activated** supplies the value "true" for all steps whose step symbol (option button or checkbox) are displayed in white or green **with** checkmark and the value "false" for all other step symbols without a checkmark.

The step is activated now, but it is not considered during package execution as long as the step is additionally assigned the "Execute" status and displayed in **green**.

Example activate step



2.2 SINUMERIK Operate auf PCU

Step	
Name	SINUMERIK Operate au
Components	Number of entries = 1
ID	7
Step configuration	
Activation	Yes
Execute	No
Activation disabled	Yes
Collapse	No
Miscellaneous	
Information text	
Warning message	None
Step scripts	
Scripts	Number of scripts = 1

Name
Defines the name of the step. This can be freely significance.

Name	Aktionstyp	Be
☑ Kopieren	Copy	fro

Figure 4-82 Display activate step

The step configuration is configured for each step individually, either via the context menu or via the settings of the property field.

Note

Step activation

The step activation can also be configured direct in the step tree and becomes effective immediately.

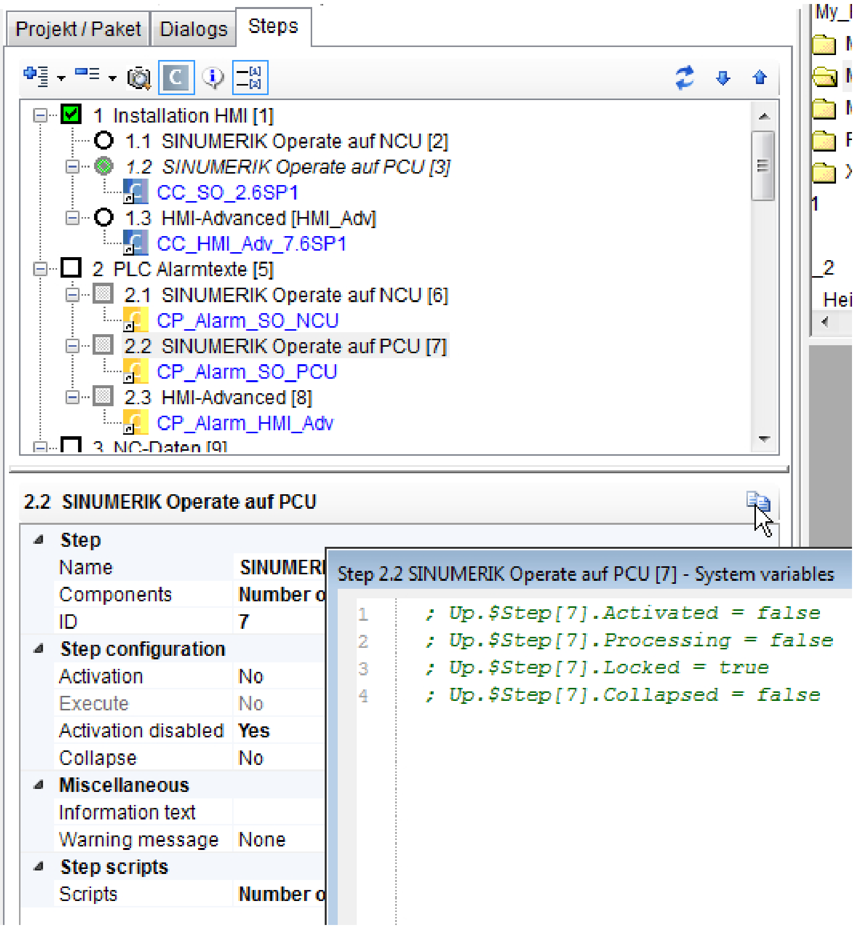
Execute step(s) - Up.$Step[id].Processing

Using the system variable `Up.$Step[id].Processing`, the execution status for package execution is defined in the step tree.

The system variable **Up.$Step[id].Processing** supplies the value "true" for all steps whose step symbol (option button or checkbox) has already been activated and are displayed in green. The selected step in green and all of its higher-level steps are performed during package execution. Activated substeps are marked in green and are thus executed if the higher-level step is activated. The step in the first level is always assigned the execution status during activation.

The value "false" is returned for all other step symbols with a white background.

Example execute step



2.2 SINUMERIK Operate auf PCU

Step	
Name	SINUMERI
Components	Number o
ID	7
Step configuration	
Activation	No
Execute	No
Activation disabled	Yes
Collapse	No
Miscellaneous	
Information text	
Warning message	None
Step scripts	
Scripts	Number o

Step 2.2 SINUMERIK Operate auf PCU [7] - System variables

```
; Up.$Step[7].Activated = false
; Up.$Step[7].Processing = false
; Up.$Step[7].Locked = true
; Up.$Step[7].Collapsed = false
```

Figure 4-83 Display execute step

Via the button (Display all system variables via a dialog), you can display the current status of the system variables for each step.

Note

Executing steps

The "Execute" status can also be configured directly in the step tree and becomes effective immediately.

Step activation locked - Up.$Step[id].Locked

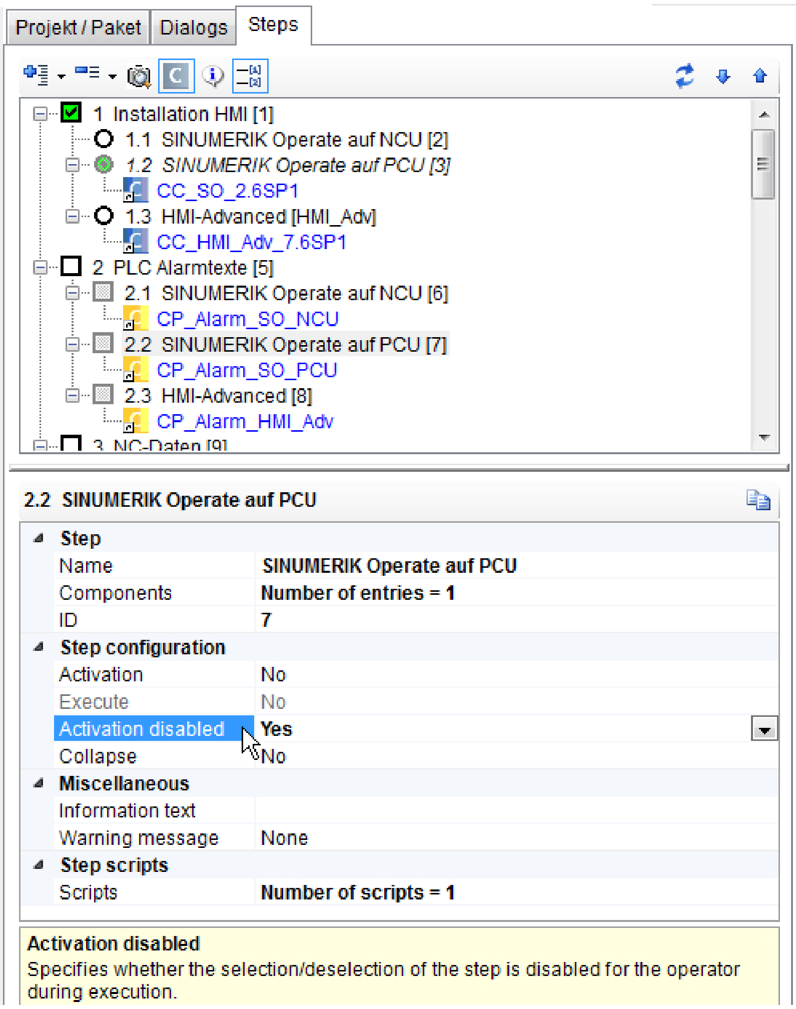
Using the system variable `Up.$Step[id].Locked`, the activation status of a step is locked for package execution.

The system variable **Up.$Step[id].Locked** supplies the value "true" for all steps whose step symbol (option button or checkbox) is displayed with a gray background, including the step. This step is not performed during package execution.

Observe the type of blocking during step configuration:

- Activation disabled
- Activation disabled including substeps

Example activation disabled



☑ 1 Installation HMI [1]
- ○ 1.1 SINUMERIK Operate auf NCU [2]
- ● *1.2 SINUMERIK Operate auf PCU [3]*
  - CC_SO_2.6SP1
- ○ 1.3 HMI-Advanced [HMI_Adv]
  - CC_HMI_Adv_7.6SP1

□ 2 PLC Alarmtexte [5]
- ▨ 2.1 SINUMERIK Operate auf NCU [6]
  - CP_Alarm_SO_NCU
- ▨ 2.2 SINUMERIK Operate auf PCU [7]
  - CP_Alarm_SO_PCU
- ▨ 2.3 HMI-Advanced [8]
  - CP_Alarm_HMI_Adv

□ 3 NC-Daten [9]

2.2 SINUMERIK Operate auf PCU

Step	
Name	**SINUMERIK Operate auf PCU**
Components	**Number of entries = 1**
ID	**7**
Step configuration	
Activation	No
Execute	No
Activation disabled	**Yes**
Collapse	No
Miscellaneous	
Information text	
Warning message	None
Step scripts	
Scripts	**Number of scripts = 1**

Activation disabled
Specifies whether the selection/deselection of the step is disabled for the operator during execution.

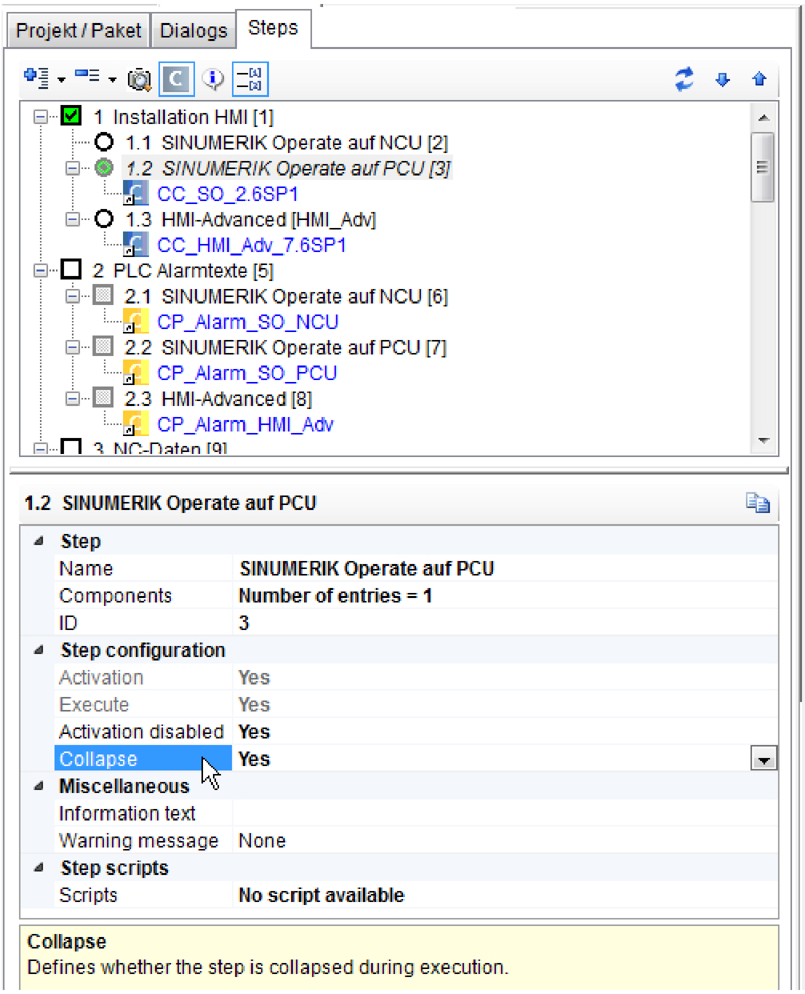
Figure 4-84 View step - Activation disabled

Collapse step(s) - Up.$Step[id].Collapsed

Using the system variable `Up.$Step[id].Collapsed`, the selected step is collapsed in the step tree. The operator cannot see the step.

The system variable **Up.$Step[id].Collapsed** supplies the value "true" for all steps whose font of the step is italic.

Example collapsing steps



- 1 Installation HMI [1]
  - 1.1 SINUMERIK Operate auf NCU [2]
  - *1.2 SINUMERIK Operate auf PCU [3]*
    - CC_SO_2.6SP1
  - 1.3 HMI-Advanced [HMI_Adv]
    - CC_HMI_Adv_7.6SP1
- 2 PLC Alarmtexte [5]
  - 2.1 SINUMERIK Operate auf NCU [6]
    - CP_Alarm_SO_NCU
  - 2.2 SINUMERIK Operate auf PCU [7]
    - CP_Alarm_SO_PCU
  - 2.3 HMI-Advanced [8]
    - CP_Alarm_HMI_Adv
- 3 NC-Daten [9]

1.2 SINUMERIK Operate auf PCU

Step	
Name	**SINUMERIK Operate auf PCU**
Components	**Number of entries = 1**
ID	3
Step configuration	
Activation	Yes
Execute	Yes
Activation disabled	**Yes**
Collapse	**Yes**
Miscellaneous	
Information text	
Warning message	None
Step scripts	
Scripts	**No script available**

Collapse
Defines whether the step is collapsed during execution.

Figure 4-85 Display collapse step

Note

Collapse steps

When configuring in the step tree, you can continue to have all steps with "Collapsed" status displayed. The steps are identified by the italic font.

The "Collapsed" status only becomes effective during package execution in Shield. The user cannot have these steps displayed.

4.14.5 Environment variables

The environment variables for Expert are listed below. Environment variables are only read accessed (read only).

The enumerations are identified with the symbol ▣, the elements of the enumerations with ▤.

4.14.5.1 Enumeration, runtime environments - runtimes

Range	Variable/enumeration	Type	Description	Example
▣	**Enumeration runtime environments - Runtimes**			
	▤ Up.$Env.RunTimes.LINUX	Enumerator value	Executing the package under LINUX	
	▤ Up.$Env.RunTimes.WINDOWS	Enumerator value	Executing the package under WINDOWS	

4.14.5.2 Environment variables

Range	Variable/enumeration	Type	Description	Example
	Up.$Env.RunTime	Enumeration RunTimes	RunTimes (Page 289)	
	Up.$Env.CFID	STRING	CF card ID (string from hwversion.xml)	"2009123709B2A62AA61A"
	Up.$Env.NCU	STRING	Type from NCU (string from hwversion.xml)	"SIEMENS SINUMERIK 840D sl NCU730.3 PN"
	Up.$Env.PLC	STRING	Type from PLC (string from hwversion.xml)	"SIEMENS SINUMERIK PLC317-3PN/DP"

hwversions.xml on CF card under user/system/etc

Package execution

5

5.1 Package execution

Information about projects which have been created in Expert, which are forwarded as Linux or Windows packages, is provided under Create MyConfig packages (Page 246).

Create MyConfig - Topo 6

6.1 Safety instructions

6.1.1 Fundamental safety instructions

6.1.1.1 General safety instructions

⚠ WARNING

Risk of death if the safety instructions and remaining risks are not carefully observed

If the safety instructions and residual risks are not observed in the associated hardware documentation, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

⚠ WARNING

Danger to life or malfunctions of the machine as a result of incorrect or changed parameterization

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

- Protect the parameterization (parameter assignments) against unauthorized access.
- Respond to possible malfunctions by applying suitable measures (e.g. EMERGENCY STOP or EMERGENCY OFF).

6.1.1.2 Industrial security

Note

Industrial security

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

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

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

⚠ **WARNING**

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

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

- Keep the software up to date.
  You will find relevant information and newsletters at this address (http://support.automation.siemens.com).
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
  You will find further information at this address (http://www.siemens.com/industrialsecurity).
- Make sure that you include all installed products into the holistic industrial security concept.

6.2 Introduction

Note

Setup of documentation: Operating Manual / Online Help

The contents for the section **Topo** are identical 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.

Comparison topology

An image of the SINAMICS topology can be used, for example, to check the topology (wiring) of the SINAMICS on the machine, to monitor module/component properties, to change the module/component names, to change DO numbers and DO names and to assign DO variables to the SINAMICS modules. SINAMICS modules or components are, for example, sensor and motor modules.

User-specified topology

In addition, it is also possible instead of the use of the automatic configuration to specify a **S**INAMICS **O**ffline (reference) **T**opology package (**SOT**). This variant of the device commissioning supports a step-by-step production, because modules/components and data can be introduced into the controller that physically do not exist.

During series production of modular machines, generally a special version of the axis-drive assignment is specified for each machine variant to be manufactured. The objective is to split the axis-drive assignment into several modular objects to be able to handle their parameterization flexibly, irrespective of the physical design and wiring of the machine. The functional interaction of the individual axis/drive configurations must be ensured beforehand.

Extended user-specified topology

As a new function in Topo - Create MyConfig V4.7 or higher, the extended user-specified topology is available in addition to the comparison and user-specified topology. In addition to the SINAMICS wiring diagram, it also contains NC and DRV data.

The starting basis for the extended user-specified topology is the archive import of a functioning machine. After changing the topology or NC or DRV data, the result can be loaded directly in NC.

An Expert project with subsequent Shield package in not required in this case.

Functions of Topo

Topo provides the following functions that are necessary for:

- Creation of topologies via a graphical user interface using the context menu and additional standard Windows tools.
- Import of topologies from SINUMERIK archives, from deployed topologies (ECAD XML files)
- Verification of package for created topologies
- Comparison of topologies
- Assignment of module/component properties such as component name, OEM ID, etc.
- Specification of criteria (Article No.), which the package compares with the actual properties of the SINAMICS modules on the machine to be produced (read-out TARGET topology of the SINAMICS), and reports any impermissible deviations.
- Assignment of the DO variables to modules/components.
- Creation of a user-defined topology based on generic modules. This user-defined topology can be downloaded directly to the SINAMICS when executing the package.

- Support of a step-by-step commissioning for missing components through deactivation of these components.
- Use the axis-drive assignment (AAZ) to assign the configured axes to the drives that are actually being used.

Assignment of DO variables and DO properties

DO variables can be assigned to the modules in Topo. These DO variables provide properties that can be used for the configuration in Expert when executing the package. In this way, it is possible to work flexibly in Expert with defined DO variables whose absolute values only apply during the package runtime on the machine.

SINAMICS modules that do not generate their own DO can be assigned the DO-generating modules via DO variables (e.g. Sensor Module to Motor Module).

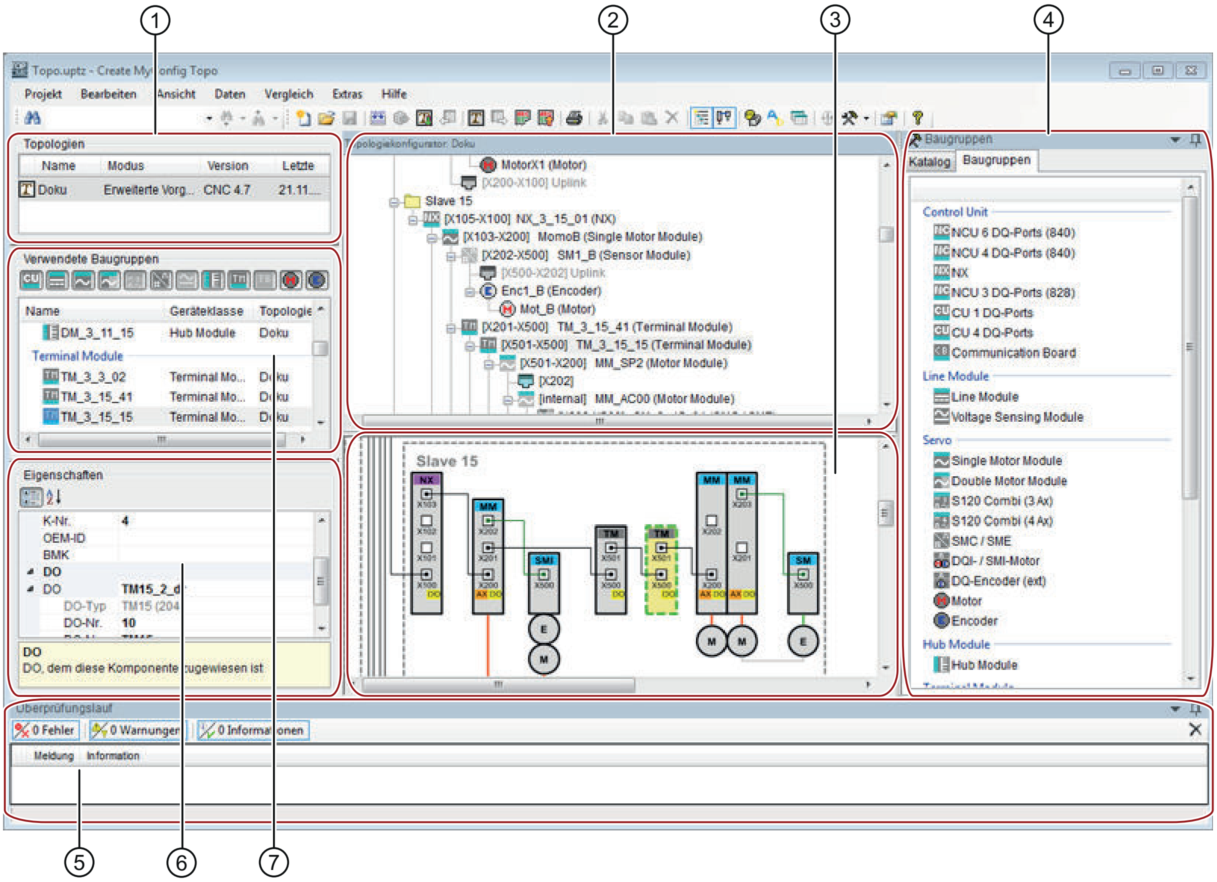
The drive-specific parameters can therefore be stored modularly in a package without reference to a particular DO number, module number, or slave. The assignment of the parameters to the topology-specific DOs on the machine is controlled flexibly via the DO variables, manipulation tasks and step scripts defined in the configuration of the package.

The DRIVE-CLiQ wiring created in Topo and the assignments made for DO variables, DO names and DO numbers are saved via a deploy function to a comparison (*.ust) or user-defined topology file (*.utz) which is accessed when a package is executed.

6.3 Overview of user interface

6.3.1 Topo - overview window

The following shows the various operating areas of the Topo software user interface:



①	"Topologies"	
②	Topology tree	Topology tree/topology graph: Both of these display forms 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 types. The topology tree or the topology graph are shown or hidden via the menu "View > Topology tree" and "View > Topology graph" .
③	Topology graph	
④	"Modules"	
⑤	"Verification of package"	
⑥	"Properties"	
⑦	"Modules used"	

6.3.2 Topo - menu functions

Designation			Meaning
Project			
	New	<Ctrl+N>	Create a new project.
	Open...	<Ctrl+O>	Open an existing project (*.uptz).
	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 symbol is active so that the changes can be saved.
	Save as...		Saves the current project under another path or file name.
	Verification of package		Checks an opened topology regarding the correct allocation of DO variables, component numbers, errors in the axis-drive assignment (AAZ). A check is not made as to whether the wiring is permissible according to SINAMICS guidelines.
	Deployment		Deploys an open topology in a .ust (reference topology) or .utz file (target topology), which can be used by the package. The topology is first checked.
	Load topology in controller		The finished project can be loaded into the controller via a network connection. A configured Shield package is being executed in the background. This is not an archive. The topology can **only** be loaded into the controller with the extended user-specific topology. It must be ensured that source data (archives or ASCI datasets) is not generated from a software version that is newer than the target software version.
	Displaying/importing topology		Displays a previously exported topology (*.ust) created by an electrical engineering design system (*.xml), a topology extracted from a drive archive or directly from a topology taken from a control. In this case, the external topology is opened, write-protected and is not imported into the actual project. When closing the topology, this can be imported into the project.
		From topology...	
		From archive...	
		From control	For this purpose, a network connection must be established, specifying the IP address.
	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		The projects opened last can be selected.
	Displayed last		The projects displayed last can be selected.
	Close all		Closes the open project.
	Exit	<Alt+F4>	Ends the Topo program.
Edit			
	Cut	<Ctrl+X>	Removes the selected components and copies them to the clipboard.
	Copy	<Ctrl+C>	Copies the selected components to the clipboard.

Designation			Meaning
	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.
View			
	Topology tree		Shows or hides the topology tree. The topology tree and graphic representation can never be hidden simultaneously.
	Topology graph		Shows or hides the graphic topology view. The topology tree and graphic representation can never be hidden simultaneously.
	Display modules used		For the various device classes, shows group titles for the modules used or hides these.
	Groups in modules used		For the various device classes, shows group titles in the "Modules used" window or hides these.
	Toolbars	Standard	Allows toolbars to be hidden or displayed.
		Quick search	
	Go to error/warning forward	<F8>	Errors or warnings can be displayed in succession.
	Go to error/warning backward	<Shift+F8>	Preceding errors or warnings can be displayed.
Data			
	Manage DO list...	Ctrl+Alt+D	Opens the "DO list" dialog for the administration of DO variables, i.e. DO variables can be newly added, imported, exported, or linked. You can find additional information under Manage DO list (Page 307).
AD	Administrate axis-drive assignment...	Ctrl+Alt+A	Opens the "Axis-drive assignment - Topo project" dialog, in which the axis-drive assignment table is shown and an axis-drive assignment 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. It is possible to perform the desired axis-drive assignment manually. You can find additional information under Axis-drive assignment (Page 327).
	Manage NC data...	Ctrl+Alt+N	Opens the "NC data - Topo project" dialog. The view of the dialog is identical to the view under Diff. Machine data or parameters are displayed for processing. Light gray parameters with the identifier "calculated" are automatically processed by Topo and cannot be changed. If the drive data is deleted, the DO is empty. It is **not** possible to add parameters.
Comparison			
	Refresh	Ctrl+D	

Designation			Meaning
	Compare with External Topology		Compares the currently displayed topology with a previously exported topology (*.ust), a topology from an electrical engineering design system (*.xml) or with a topology from a drive archive. In this case, the external topology is opened, write-protected and is not imported into the actual project. You can find additional information under Topology comparison (Page 309).
T		From topology	
		From archive	
	Edit configuration	Ctrl+M	Opens the dialog "Configuration - Topology comparison", in which you can manage the comparison configurations (i.e. create, edit, delete, etc.).
	Select configuration	Topology comparison	Compares the currently displayed topologies and switches over into the compare mode. You can find additional information under Topology comparison (Page 309). To display several topologies, select these in the list of topologies using the <Ctrl> key and a mouse click. A maximum of two topologies can be compared simultaneously.
		Complete comparison	
Tools			
	Options...		Opens the "Options" dialog, which offers the option of making basic settings, such as language and color changes to the graphical representation, in the following tabs: • Topology graph • Axis-drive assignment • Messages • Language
Help			
?	Contents...	<F1>	Calls the online help of the Topo program.
	About...		Contains information on the version and copyright.

6.4 Topo - operation

6.4.1 Creating a topology

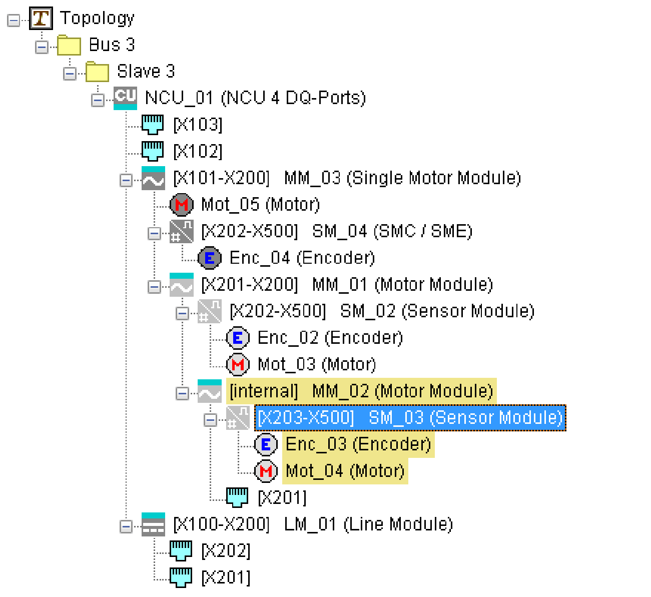
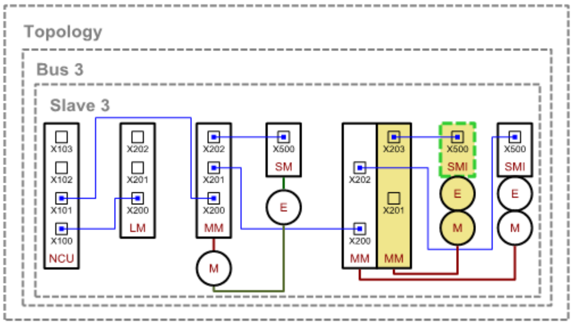
Topo offers the option of creating a new topology from the module window. Furthermore, it is possible to import and edit a topology from an electrical design system or directly from a control via an archive.

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. During the import, it can also be selected as to whether the DO properties should be taken over from the archive. In so doing, new DO variables are automatically generated and assigned to the imported components.

As from a newer CNC software version, the DO variables used for CNC commissioning are stored in the control and can be assigned and displayed again.

A topology is displayed in two ways in Topo. On the one hand, in a structurally hierarchic tree, and on the other hand, in a systematic graphic form which allows the wiring to be more easily understood and individual components to be simply identified.
Both of these display types offer the same functionality regarding creating and editing a topology. As a consequence, you can work with one or both of these display types. The topology tree or the topology graph can be shown or hidden via the menu "View > Topology Tree" and "View > Topology Graph."

To display and edit several topologies, they can be selected in the list of topologies by making a multiple selection. Just the same as for Comparing topologies (Page 309), both technologies are displayed next to one another.

Topology tree	Topology graph
 Topology Bus 3 Slave 3 NCU_01 (NCU 4 DQ-Ports) [X103] [X102] [X101-X200] MM_03 (Single Motor Module) Mot_05 (Motor) [X202-X500] SM_04 (SMC / SME) Enc_04 (Encoder) [X201-X200] MM_01 (Motor Module) [X202-X500] SM_02 (Sensor Module) Enc_02 (Encoder) Mot_03 (Motor) [internal] MM_02 (Motor Module) [X203-X500] SM_03 (Sensor Module) Enc_03 (Encoder) Mot_04 (Motor) [X201] [X100-X200] LM_01 (Line Module) [X202] [X201]	 Topology Bus 3 Slave 3 X103 X102 X101 X100 NCU X202 X201 X200 LM X202 X201 X200 MM M X500 SM E X202 X200 MM X203 X201 MM X500 SMI E M X500 SMI E M

Procedure for creating a topology

1. Start Topo.
2. In the "Project" main menu, select the "New" entry to create a new project.
   A new topology is now automatically created with the default name "Topology" and a "Bus 3" inserted.
3. 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 external CU, e.g. CU320

Note that these cannot be connected to the internal "Bus 3." To do this, first insert the relevant bus 1, 2, 4 ... into the topology.

4. The components are wired either from the selected structure node of the tree view or directly from the wiring diagram via the "Connect component" context menu entry.
   The components that can be used at this point of the topology can be selected from the module window and inserted and wired in the list of the used components. In this way, it is ensured that a valid topology is created and that components are not connected twice.
   Alternatively, you can drag and drop the components from the module window to the "topology configurator" and connect them (save).

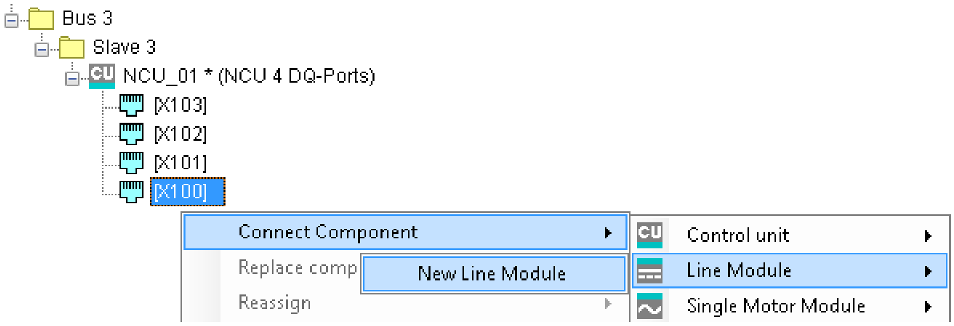
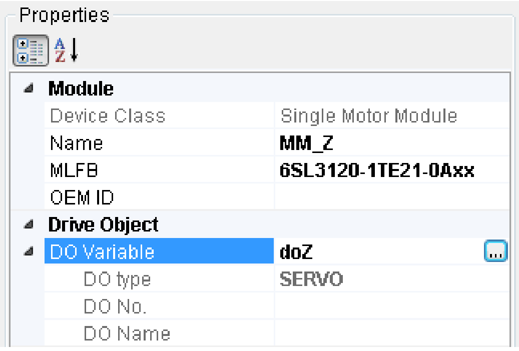


Figure 6-1 Creating a topology

5. The DO variables are assigned to the components in the component properties in the "DO variables" field. Before you can make this assignment, you must have created or imported the DO variables. Refer to the section Manage DO list (Page 307).


Properties

Module	
Device Class	Single Motor Module
Name	MM_Z
MLFB	6SL3120-1TE21-0Axx
OEM ID	
Drive Object	
DO Variable	doZ
DO type	SERVO
DO No.	
DO Name	

Figure 6-2 Assigning DO variables

Additional functions

In addition to the standard functions such as "Cut", "Copy", "Paste", etc., the context menu of the topology configurator provides further functions:

Function	Description
Edit	If a topology has been read in via an archive, it will be displayed write-protected on a gray background. By editing or by clicking on the Import button, it is included in the topology list and can be processed.
Replace component	Replaces an already connected component with an equivalent component from a different topology of the opened topology project. Only components from the same device class and that have the same number of DRIVE-CLiQ ports are listed. When replacing components, all of the components wired to the original component are kept.

Function	Description
Convert Component	Converts the selected component and directly connected lower-level components into another module. SMI, SMC and DQE modules can be converted in this way.
Rewire	**Downlink port**: In Topo, a component is always generally wired to the lower-level component with its first DRIVE-CLiQ port (X100, X200, ...). If the wiring is to be made to another port, then any free DRIVE-CLiQ port can be selected here.
	Uplink port:. In Topo, as a standard, a component is always wired to the higher-level component using its first DRIVE-CLiQ port (X100, X200, etc.). If the wiring is to be made to another port, then any free DRIVE-CLiQ port can be selected here. Rewiring is also possible by means of drag-and-drop
Search	Lets you search for character strings within the topology tree.
Print	Allows a topology tree or a topology graph to be printed out. For a topology graph, in addition, a legend of all properties can also be printed out at the same time.
Zoom	**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. The size can also be changed as required by holding down "Ctrl" and turning the mouse wheel.

See also

Managing modules (Page 303)

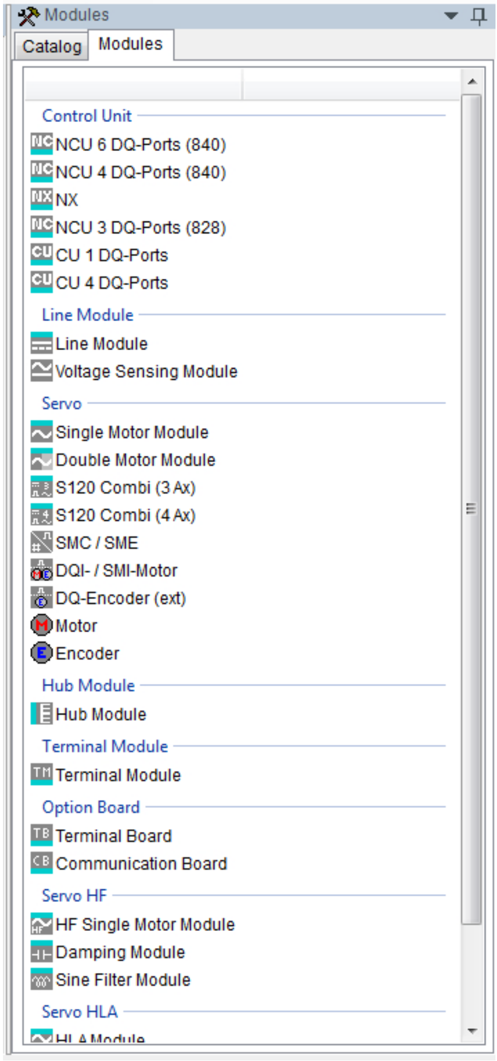
Topo - menu functions (Page 298)

6.4.2 Managing modules

Modules

The module window integrated in Topo contains templates of modules for each supported device class. The module window is structured according to specific groups of topics.

These modules can be dragged to the topology configurator and connected to a free DRIVE-CLiQ port.



Control Unit
- NCU 6 DQ-Ports (840)
- NCU 4 DQ-Ports (840)
- NX
- NCU 3 DQ-Ports (828)
- CU 1 DQ-Ports
- CU 4 DQ-Ports

Line Module
- Line Module
- Voltage Sensing Module

Servo
- Single Motor Module
- Double Motor Module
- S120 Combi (3 Ax)
- S120 Combi (4 Ax)
- SMC / SME
- DQI- / SMI-Motor
- DQ-Encoder (ext)
- Motor
- Encoder

Hub Module
- Hub Module

Terminal Module
- Terminal Module

Option Board
- Terminal Board
- Communication Board

Servo HF
- HF Single Motor Module
- Damping Module
- Sine Filter Module

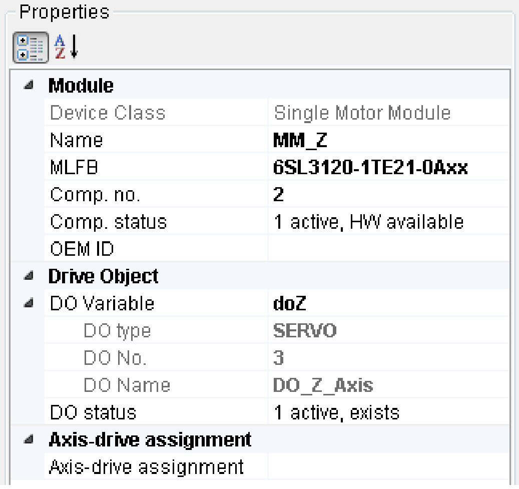
Servo HLA
- HLA Module

Figure 6-3 Modules

Component properties

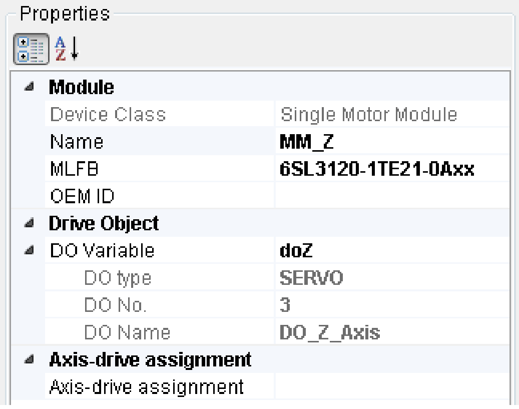
Each component can be assigned various properties, depending on the device class:

Component properties of user-defined topology (*.utz)



Module	
Device Class	Single Motor Module
Name	**MM_Z**
MLFB	**6SL3120-1TE21-0Axx**
Comp. no.	**2**
Comp. status	1 active, HW available
OEM ID	
Drive Object	
DO Variable	**doZ**
DO type	**SERVO**
DO No.	**3**
DO Name	**DO_Z_Axis**
DO status	1 active, exists
Axis-drive assignment	
Axis-drive assignment	

Component properties of comparison topology (*.ust)



Module	
Device Class	Single Motor Module
Name	**MM_Z**
MLFB	**6SL3120-1TE21-0Axx**
OEM ID	
Drive Object	
DO Variable	**doZ**
DO type	**SERVO**
DO No.	**3**
DO Name	**DO_Z_Axis**
Axis-drive assignment	
Axis-drive assignment	

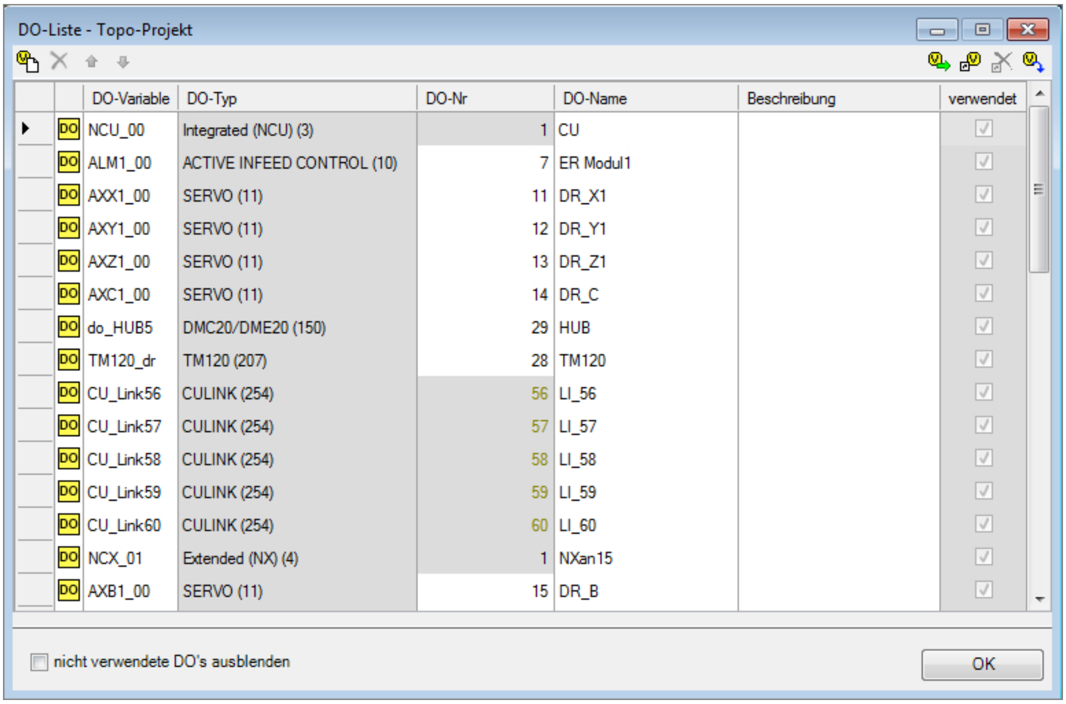
Property	Description	Note
Component group		
Name	Name of the component.	Component names must be unique and have a maximum of 50 ASCII characters.
Article no.	Order number of the component.	You can specify the article number so that in addition to the wiring, the package also checks the use of the correct components 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 when executing the package, the package generates an alarm that the operator can ignore. If a "?" is in a position, this position will be specified by the SINAMICS on importing the user-defined topology.

Property	Description	Note
K 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 is between 1 and 199. • Component number 1 is assigned automatically for control units and cannot be edited. • In relation to a control unit, the component number must be unique. • For all other components, the component number can be in the range from 2 to 193 in relation to a control unit (CU320, NCU or NX) (6 K numbers are reserved for CU-LINK DOs). • In the topology configurator, the component numbers can be automatically reassigned or added via the context menu when selecting a bus or slave.
K status	Setting options: • HW available, inactive (p1x5=0) • HW available, active (p1x5=1) • HW not available, inactive (p1x5=2)	• The default value is "HW available, active." • Cannot be changed for CU320 and NCU control units. • Also cannot be changed for DOs that only consist of one component.
OEM ID	Identifier of the component in the catalog of the machine manufacturer.	Only numbers can be used.
DO status		
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 (see Topo - menu functions (Page 298)) that are permitted on this component in accordance with their type.	Not available 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.
DO type	Assignment of a DO to a component type.	Is a property of the DO variable and can only be modified via the DO variable list (Page 307).
DO number	DO number via which a drive object is addressed.	Is a property of the DO variable and can only be modified via the DO variable list (Page 307).
DO name	Freely selectable name for a drive object.	Is a property of the DO variable and can only be modified via the DO variable list (Page 307). DO names can have a maximum of 24 ASCII characters.
DO status	Setting options: • 0 inactive, available • 1 active, available • 2 not available	• The default value is "1 active, available". • The "Active" status cannot be changed for the DO types "Integrated (NCU)" and "SINAMICS S".
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 device class "Sensor Module".

6.4.3 Manage DO list

The DO list that you can call via the main menu "Data > Manage DO list" is used to manage the DO variables used in the topology project.
DO variables can either be created manually or imported from an existing UVD file and then edited. The UVD file serves as an interface between the Expert and Topo configuring tools.
DO variables can be exported from Topo as well as from Expert in a UVD file () or imported from this (). It is also possible to link a UVD file ().

We recommend linking the same DO list in Topo and Expert.



DO-Liste - Topo-Projekt

	DO-Variable	DO-Typ	DO-Nr	DO-Name	Beschreibung	verwendet
▶	DO NCU_00	Integrated (NCU) (3)	1	CU		☑
	DO ALM1_00	ACTIVE INFEED CONTROL (10)	7	ER Modul1		☑
	DO AXX1_00	SERVO (11)	11	DR_X1		☑
	DO AXY1_00	SERVO (11)	12	DR_Y1		☑
	DO AXZ1_00	SERVO (11)	13	DR_Z1		☑
	DO AXC1_00	SERVO (11)	14	DR_C		☑
	DO do_HUB5	DMC20/DME20 (150)	29	HUB		☑
	DO TM120_dr	TM120 (207)	28	TM120		☑
	DO CU_Link56	CULINK (254)	56	LI_56		☑
	DO CU_Link57	CULINK (254)	57	LI_57		☑
	DO CU_Link58	CULINK (254)	58	LI_58		☑
	DO CU_Link59	CULINK (254)	59	LI_59		☑
	DO CU_Link60	CULINK (254)	60	LI_60		☑
	DO NCX_01	Extended (NX) (4)	1	NXan15		☑
	DO AXB1_00	SERVO (11)	15	DR_B		☑

□ nicht verwendete DO's ausblenden

Figure 6-4 Manage DO list

Assigning properties to variables

You can assign the following properties to each DO variable:

Property	Description	Note
DO variable	Name of the DO variable used in the package for assignment to a drive object.	Valid characters are: a-z, A-Z, 1-9, _ (no numbers at the beginning, min. 2, max. 24 characters)
DO type	Defining the DO type.	-

Property	Description	Note
DO No.	DO number via which a drive object is addressed.	Only numbers between 2 and 54 as well as 61 and 62 are permitted. For Control Units (CU_I, NX) only a "1" is permissible. The DO number 63 is reserved for DO number-independent BICO connections within a DO. The numbers 55 to 60 are reserved for "CU-LINK" DOs.
DO name	Freely selectable name for a drive object.	Max. 25 characters.
Description	Freely selectable description text for the DO variable.	-

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	Servo, Servo HF
Determine TM from HW TM31 TM41 TM15 TM120	Terminal module
TB30	Terminal Board
DMC20 / DME20	Hub Module
CU-LINK	-

Number of components

Restrictions with regard to the number of components that can be assigned a DO also apply. Only the following assignments are permitted:

DO type	Component type	Permitted number
SINAMICS S (CU320)	CU320, all variants	1
Integrated (NCU)	NCU, all variants	1
Extended (NX)	NX, all variants	1
	Line module	1
	Voltage sensor module	1

DO type	Component type	Permitted number
	Motor module	1
	Motor	1
	Sensor module	2
	Encoder	2
	Terminal Module	1
	Terminal Board	1
	Hub Module	2
	NX10 or NX15	1

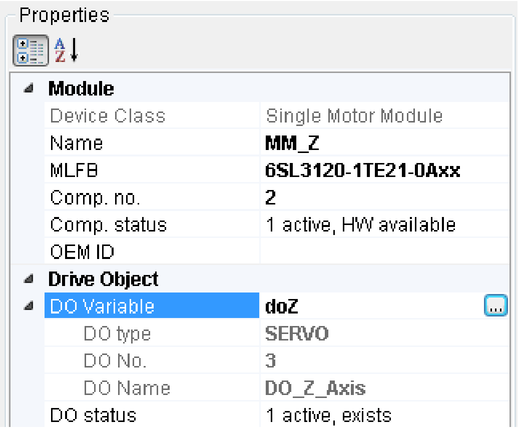
Note

You can recognize which components are assigned to which DOs.

If you select a wired component, the remaining components that are assigned to the same DO are highlighted.

Display of the DO variables

The properties of the DO variables assigned to a component are displayed in the "Properties" dialog box.



Properties

Module	
Device Class	Single Motor Module
Name	**MM_Z**
MLFB	**6SL3120-1TE21-0Axx**
Comp. no.	**2**
Comp. status	1 active, HW available
OEM ID	
Drive Object	
DO Variable	**doZ**
DO type	**SERVO**
DO No.	**3**
DO Name	**DO_Z_Axis**
DO status	1 active, exists

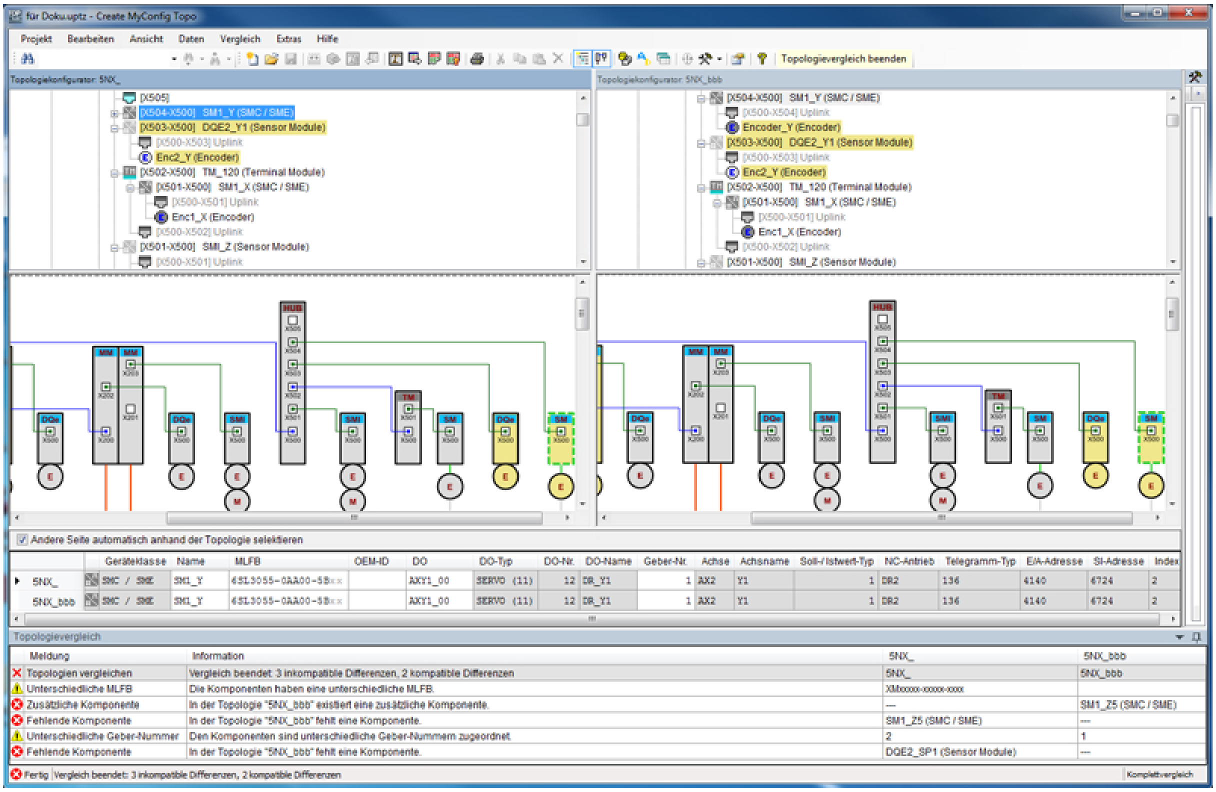
Figure 6-5 Component properties

6.4.4 Comparing topologies

The topology comparison offers the option of comparing two topologies as well as components - or to synchronize the individual properties of a component.

Two topologies in the current project that have been selected can be compared via the menu "Compare > Refresh" .

It is also possible to compare a topology in the project with one from an archive or from a .ust or .utz file via the menu "Comparison > Compare with External Topology".


für Doku.uptz - Create MyConfig Topo
Projekt
Bearbeiten
Ansicht
Daten
Vergleich
Extras
Hilfe
Topologievergleich beenden
Topologiekonfigurator: 5NX_
Topologiekonfigurator: 5NX_bbb
HUB
MM
DQe
SMI
SM
TM
E
M
X500
X501
X502
X503
X504
X505
X200
X201
X202
X203

Topologiekonfigurator: 5NX_

```
[X505]
[X504-X500]  SM1_Y (SMC / SME)
[X503-X500]  DQE2_Y1 (Sensor Module)
    [X500-X503] Uplink
    Enc2_Y (Encoder)
[X502-X500]  TM_120 (Terminal Module)
    [X501-X500]  SM1_X (SMC / SME)
        [X500-X501] Uplink
        Enc1_X (Encoder)
    [X500-X502] Uplink
[X501-X500]  SMI_Z (Sensor Module)
    [X500-X501] Uplink
```

Topologiekonfigurator: 5NX_bbb

```
[X504-X500]  SM1_Y (SMC / SME)
    [X500-X504] Uplink
    Encoder_Y (Encoder)
[X503-X500]  DQE2_Y1 (Sensor Module)
    [X500-X503] Uplink
    Enc2_Y (Encoder)
[X502-X500]  TM_120 (Terminal Module)
    [X501-X500]  SM1_X (SMC / SME)
        [X500-X501] Uplink
        Enc1_X (Encoder)
    [X500-X502] Uplink
[X501-X500]  SMI_Z (Sensor Module)
```

☑ Andere Seite automatisch anhand der Topologie selektieren

	Geräteklasse	Name	MLFB	OEM-ID	DO	DO-Typ	DO-Nr.	DO-Name	Geber-Nr.	Achse	Achsname	Soll-/Istwert-Typ	NC-Antrieb	Telegramm-Typ	E/A-Adresse	SI-Adresse	Index
5NX_	SMC / SME	SM1_Y	6SL3055-0AA00-5Bxx		AXY1_00	SERVO (11)	12	DR_Y1	1	AX2	Y1	1	DR2	136	4140	6724	2
5NX_bbb	SMC / SME	SM1_Y	6SL3055-0AA00-5Bxx		AXY1_00	SERVO (11)	12	DR_Y1	1	AX2	Y1	1	DR2	136	4140	6724	2

Topologievergleich

Meldung	Information	5NX_	5NX_bbb
✗ Topologien vergleichen	Vergleich beendet: 3 inkompatible Differenzen, 2 kompatible Differenzen	5NX_	5NX_bbb
⚠ Unterschiedliche MLFB	Die Komponenten haben eine unterschiedliche MLFB.	XMxxxxx-xxxxx-xxxx	
⊗ Zusätzliche Komponente	In der Topologie "5NX_bbb" existiert eine zusätzliche Komponente.	---	SM1_Z5 (SMC / SME)
⊗ Fehlende Komponente	In der Topologie "5NX_bbb" fehlt eine Komponente.	SM1_Z5 (SMC / SME)	---
⚠ Unterschiedliche Geber-Nummer	Den Komponenten sind unterschiedliche Geber-Nummern zugeordnet.	2	1
⊗ Fehlende Komponente	In der Topologie "5NX_bbb" fehlt eine Komponente.	DQE2_SP1 (Sensor Module)	---

⊗ Fertig | Vergleich beendet: 3 inkompatible Differenzen, 2 kompatible Differenzen | Komplettvergleich

Figure 6-6 Comparing topologies

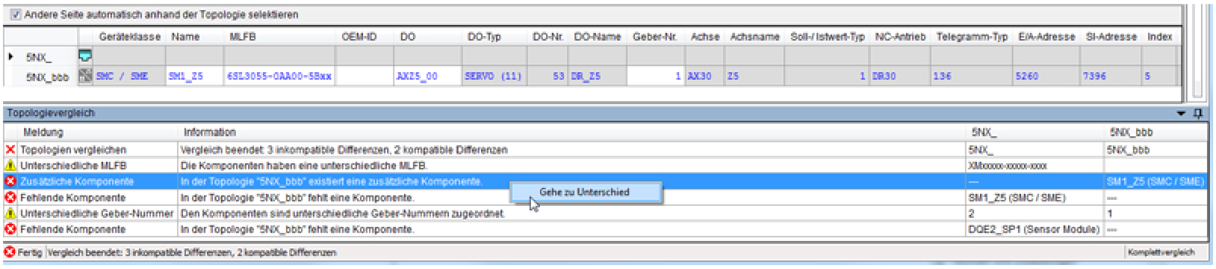
Marking the differences in the component properties

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 context 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 context menu "Go to Difference" or by double-clicking the message line.



☑ Andere Seite automatisch anhand der Topologie selektieren

	Geräteklasse	Name	MLFB	OEM-ID	DO	DO-Typ	DO-Nr.	DO-Name	Geber-Nr.	Achse	Achsname	Soll-/Istwert-Typ	NC-Antrieb	Telegramm-Typ	E/A-Adresse	SI-Adresse	Index
5NX_																	
5NX_bbb	SMC / SME	SM1_Z5	6SL3055-0AA00-5Bxx		AXZ5_00	SERVO (11)	53	DR_Z5	1	AX30	Z5	1	DR30	136	5260	7396	5

Topologievergleich

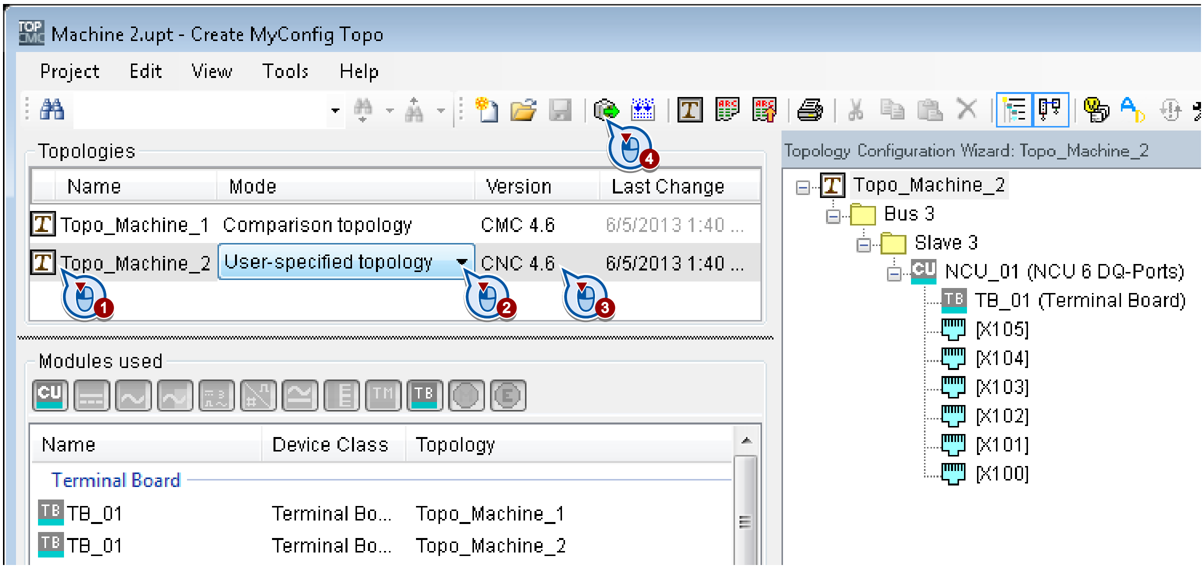
Meldung	Information	5NX_	5NX_bbb
Topologien vergleichen	Vergleich beendet: 3 inkompatible Differenzen, 2 kompatible Differenzen	5NX_	5NX_bbb
Unterschiedliche MLFB	Die Komponenten haben eine unterschiedliche MLFB.	XMxxxxx-xxxxx-xxxx	
Zusätzliche Komponente	In der Topologie "5NX_bbb" existiert eine zusätzliche Komponente.	---	SM1_Z5 (SMC / SME)
Fehlende Komponente	In der Topologie "5NX_bbb" fehlt eine Komponente.	SM1_Z5 (SMC / SME)	---
Unterschiedliche Geber-Nummer	Den Komponenten sind unterschiedliche Geber-Nummern zugeordnet.	2	1
Fehlende Komponente	In der Topologie "5NX_bbb" fehlt eine Komponente.	DQE2_SP1 (Sensor Module)	---

Fertig | Vergleich beendet: 3 inkompatible Differenzen, 2 kompatible Differenzen | Komplettvergleich

6.4.5 Deploy topology

Procedure

To deploy the topology in Topo, proceed as follows:



Topologies

Name	Mode	Version	Last Change
Topo_Machine_1	Comparison topology	CMC 4.6	6/5/2013 1:40 ...
Topo_Machine_2	User-specified topology	CNC 4.6	6/5/2013 1:40 ...

Modules used

Name	Device Class	Topology
Terminal Board		
TB_01	Terminal Bo...	Topo_Machine_1
TB_01	Terminal Bo...	Topo_Machine_2

1. Select the topology that is to be deployed.
2. Set the mode of the topology to deploy this as user-defined topology (*.utz) or comparison topology (*.ust).
3. Set the correct CNC version.
4. Start the deployment.

6.5 Command line call

Create MyConfig Topo can be executed in command line mode for the automated deployment.

Call

```
Topo <topology path>
Topo /diff <file1> <file2>
Topo /silent <topology path>
     /deploy [<deployment name>] [/deploydir <deployment directory>]
     /target (UST|UTZ27|UTZ44|UTZ45|UTZ47) [/autocomplete]
     [/log <log file>]
```

Note

Note the following rules for the command line call:

- The /silent option must always be specified first.
- Path specifications may not end with '\'.

Parameter

Options / switches	Arguments	Description
-	<Topology path>	• Specifies the path of the project file – either absolute or relative. • The path must be specified in double inverted commas if spaces are included in it. **Example**: "C:\Path with spaces\package file" • For topology projects (*.uptz), the name of the topology to be compared must be specified separately behind the path with a colon. If no name is specified, the first topology contained in the project is selected. • uptz, ust, utz, and xml files according to the Topo interface description as well as SINUMERIK archives are permitted.
/diff	<file1> <file2>	• Two paths to uptz, ust, utz, arc files. • For topology projects (*.upt), the name of the topology to be compared must be specified separately behind the path with a colon.
/silent	-	• The graphical user interface of Create MyConfig Topo is hidden. • The following options are only effective with this switch.
/deploy	<Deploy name>	• Causes the topology selected in the project to be checked and deployed. • If the parameter is omitted, Topo will generate the name from the <topology path>.
/deploydir	<Deploy directory >	• Specifies the deployment path, absolute or relative, in which the topology is to be deployed. • The path must be specified in double inverted commas if spaces are included in it. **Example**: "C:\path with spaces" • If the switch is omitted, Topo uses the folder of <topology path>.

Options / switches	Arguments	Description
/log	<Log file>	• Specifies the path of a log file – either absolute or relative. • 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.
/target	UST UTZ27 UTZ44 UTZ45 UTZ47	• Deployment type and CNC software version of the NCU on which the specified topology is used.
/autocomplete	-	• Automatically adds missing component numbers and DO properties and sets up standard encoder assignments if none have been specified.

Return values

Note

The return values for the Topo command line call only apply if Topo is executed in /silent mode without the user interface.

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	Incorrect command line call

Example

Precondition: Create MyConfig is installed in the directory "C:\Program Files\SIEMENS\Create MyConfig 4.7" and a project "TopoProj_1.upt" that contains the "Topo1" and "Topo2" topologies is available in the path "C:\Topologie."

```
Set PATH=%PATH%;"C:\Program Files\SIEMENS\Create MyConfig 4.7"

Topo.exe /silent "C:\Topologies\TopoProj_1.upt:Topo1" /deploy xyz /
deploydir "D:" /target UTZ44


if %ERRORLEVEL% EQU 3 (
@echo "command line call errors!"
pause
)
if %ERRORLEVEL% EQU 2 (
    @echo "Faults during deployment!"
    pause
)
if %ERRORLEVEL% EQU 1 (
    @echo "Alarms during deployment!"
    pause
)
if %ERRORLEVEL% EQU 0 (
    @echo "Deploy executed without error!"
    pause
)
```

6.6 SINAMICS device configuration

6.6.1 Production of modular machines

The Create MyConfig (CMC) engineering software supports both the automated production of modular machines for the machine manufacturer as well as the upgrade of machines for end customers, which requires only a limited number of operator actions.

The production of machines with 840D sl requires a basic commissioning of the SINAMICS, generally designated as "System configuration". The system configuration can be performed with CMC either as "Automatic configuration" or as "User-specified topology."

When the system configuration is configured in the CMC package, all the data sets and optimization data in the step tree required within a series of modular machines in the maximum machine configuration can be specified. 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.

6.6.2 SINAMICS device configuration with Automatic configuration or user-specified topology

Applications

A differentiation is made between three applications that CMC supports when configuring SINAMICS:

1. The SINAMICS was commissioned previously with SINUMERIK Operate or with a different commissioning software. The existing drive data can be checked with a CMC package via "Initial state" (online access) on the dialog page "System configuration" under "DRV..." in the drive archive data, including the topology. In addition, a further configuration or optimization of the SINAMICS can be made.
2. The maximum configuration of the SINAMICS is wired on 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. The 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-defined topology file is a SINAMICS offline topology package (SOT) that contains the topology in the final 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 message frames and the BICO links are also entered automatically in both CMC variants.

Table 6-1

Automatic configuration	User-specified topology
Execution via SINUMERIK Operate or CMC (identical result).	Execution only via CMC.
The 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 the automatic configuration are acquired.	The user-specified topology can also include SINAMICS modules that have not yet been wired and deactivated, exception: NCU, NX, CU320.
The DRV archive data also contains the DOs 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.

Automatic configuration	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 reports an error if the topology specified in the user-specified topology cannot be implemented.

Configuring in Expert

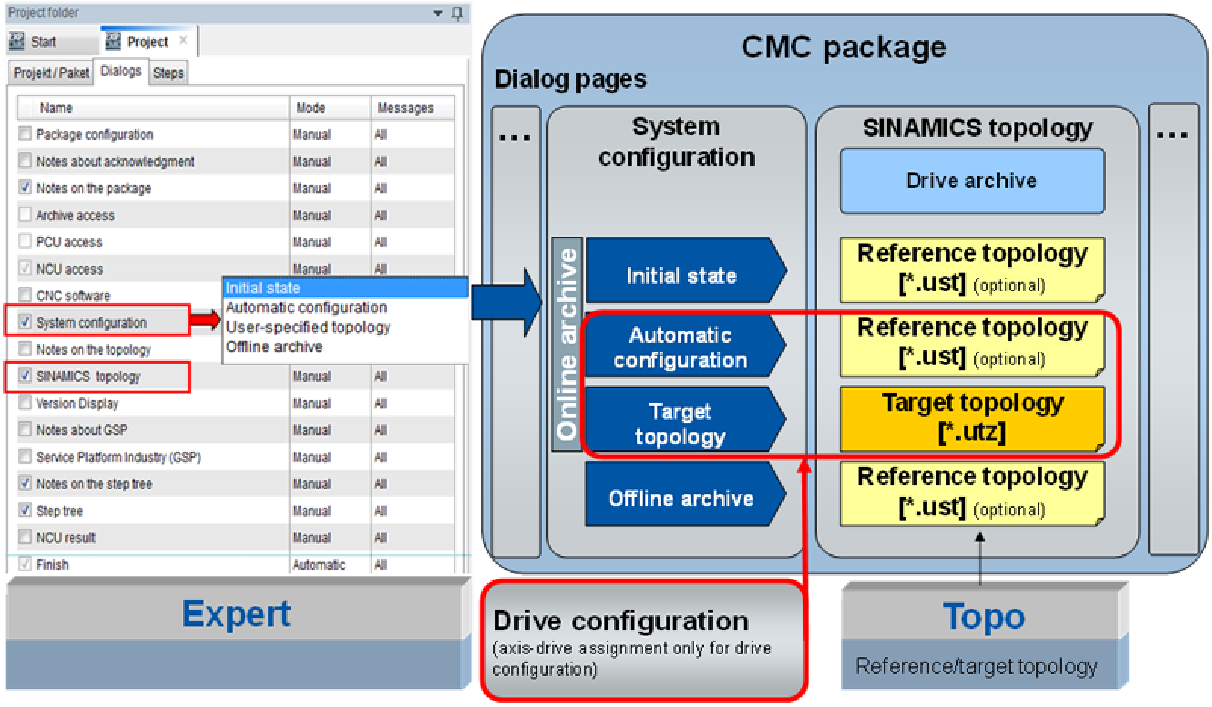
When configuring in Expert, on the "System configuration" dialog page in section "DRV..." under "Origin of data", you can define which of the four applications "Initial state," "Automatic configuration," "User-specified topology," or "Offline archive" is applicable.

By selecting the "User-specified topology" mode, it is necessary to generate with Topo a file with the user-specified topology and 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 "utz."

If "automatic configuration" is performed, as an option, you can use on the "SINAMICS topology" dialog page a file with the comparison topology in XML format and the 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 setting Origin of data" 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 and the comparison topology contain the topology of the SINAMICS and other properties of the DOs and the components.



Project folder

Start | Project

Projekt / Paket | Dialogs | Steps

Name	Mode	Messages
□ Package configuration	Manual	All
□ Notes about acknowledgment	Manual	All
☑ Notes on the package	Manual	All
□ Archive access	Manual	All
□ PCU access	Manual	All
☑ NCU access	Manual	All
□ CNC software		
☑ System configuration		
□ Notes on the topology		
☑ SINAMICS topology	Manual	All
□ Version Display	Manual	All
□ Notes about GSP	Manual	All
□ Service Platform Industry (GSP)	Manual	All
☑ Notes on the step tree	Manual	All
☑ Step tree	Manual	All
□ NCU result	Manual	All
☑ Finish	Automatic	All

Initial state
Automatic configuration
User-specified topology
Offline archive

Figure 6-7 CMC package structure

6.6.3 Drive data generation via "automatic configuration"

Basic procedure

The "automatic configuration" via SINUMERIK Operate or via CMC uses the same functions internally. In both cases, the SINAMICS determines its actual topology and saves it as target topology (p9903). This also includes the creation of drive objects (DOs) 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). The file PS000000 is used for the consistence monitoring and the file PS000099 contains configuration data of the internal PROFIBUS. This structure can be saved as drive archive and imported later.

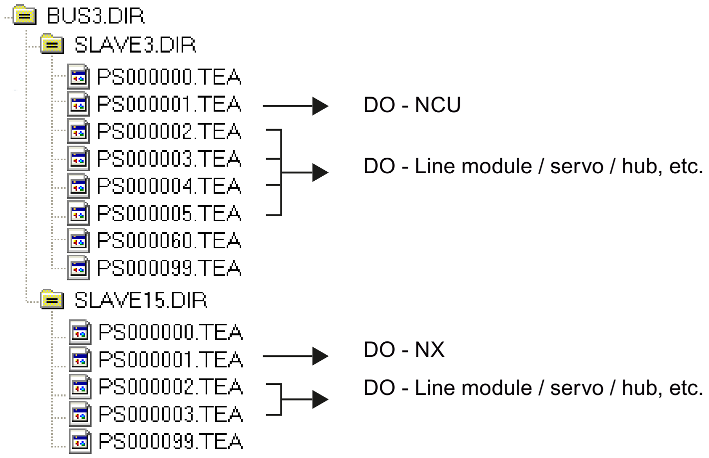


Figure 6-8 Drive data storage

SINAMICS specifies for the "Automatic configuration" DO names, DO numbers, component names, component numbers, etc., and acquires component properties such as article number, component category and component type. The parameters for the DOs are read out of the components, calculated or assigned with factory settings.

The complete data is backed up in the drive data archive.

6.6.4 Drive data generation via "user-specified topology"

Basic procedure

If the SINAMICS device configuration is made with the "User-specified topology," the SINAMICS is assigned a SINAMICS offline topology package corresponding to the maximum configuration of the machine in the form of a user-specified topology file (*.utz) from the CMC package. In other words, components, such as 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 DO names, DO numbers, component names and component numbers as well as component properties such as article number and component type. The structure with generalized drive data is also specified.

Finally, the package processing requests the SINAMICS to check this data based on the currently available configuration and to customize it appropriately, i.e. define the DO or component status. SINAMICS has in the result a customized target topology (p9903) and customized DOs for the specified components.

The DO parameters are customized by SINAMICS with the factory settings, values exported from the components or calculated values. All this data is backed up in the drive data archive.

Because at this time components not yet present in the structure have already been created, the associated user data can be added to the step tree using manipulation tasks. As an alternative, for example with DQI/SMI motors, export of the data from the electronic rating plate with the motor type or encoder type selection can be forced after activation.

6.6.5 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.

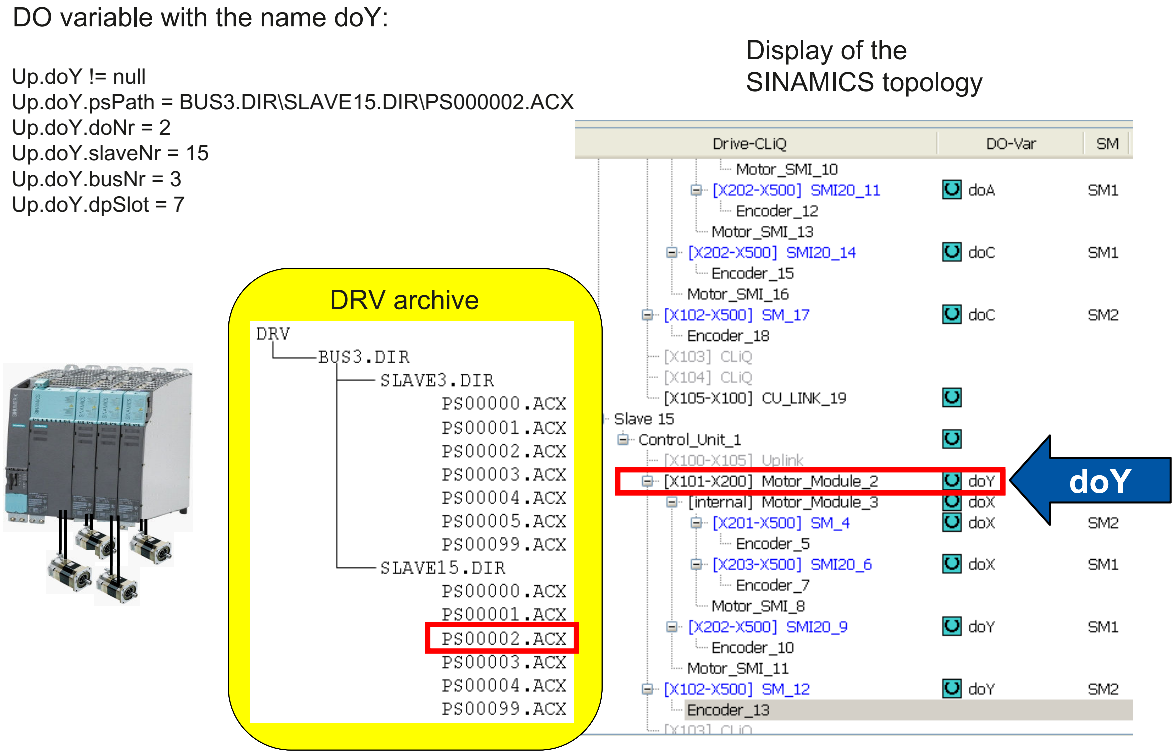
Properties of DO variables

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 6-2 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 manipulation tasks and step scripts.



DO variable with the name doY:

```
Up.doY != null
Up.doY.psPath = BUS3.DIR\SLAVE15.DIR\PS000002.ACX
Up.doY.doNr = 2
Up.doY.slaveNr = 15
Up.doY.busNr = 3
Up.doY.dpSlot = 7
```

Display of the SINAMICS topology

DRV archive

```
DRV
└──BUS3.DIR
    ├──SLAVE3.DIR
    │       PS00000.ACX
    │       PS00001.ACX
    │       PS00002.ACX
    │       PS00003.ACX
    │       PS00004.ACX
    │       PS00005.ACX
    │       PS00099.ACX
    └──SLAVE15.DIR
            PS00000.ACX
            PS00001.ACX
            PS00002.ACX
            PS00003.ACX
            PS00004.ACX
            PS00099.ACX
```

Drive-CLiQ	DO-Var	SM
Motor_SMI_10		
[X202-X500] SMI20_11	doA	SM1
Encoder_12		
Motor_SMI_13		
[X202-X500] SMI20_14	doC	SM1
Encoder_15		
Motor_SMI_16		
[X102-X500] SM_17	doC	SM2
Encoder_18		
[X103] CLiQ		
[X104] CLiQ		
[X105-X100] CU_LINK_19		
Slave 15		
Control_Unit_1		
[X100-X105] Uplink		
[X101-X200] Motor_Module_2	doY	
[internal] Motor_Module_3	doX	
[X201-X500] SM_4	doX	SM2
Encoder_5		
[X203-X500] SMI20_6	doX	SM1
Encoder_7		
Motor_SMI_8		
[X202-X500] SMI20_9	doY	SM1
Encoder_10		
Motor_SMI_11		
[X102-X500] SM_12	doY	SM2
Encoder_13		
[X103] CLiQ		

Figure 6-9 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 two ways of assigning the DO variables to the SINAMICS components:

1. When processing the package, you need to 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.

6.6.6 The advantages of using comparison or user-specified topology

The configuration of a comparison or user-specified topology withTopo 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, context menu, verification 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

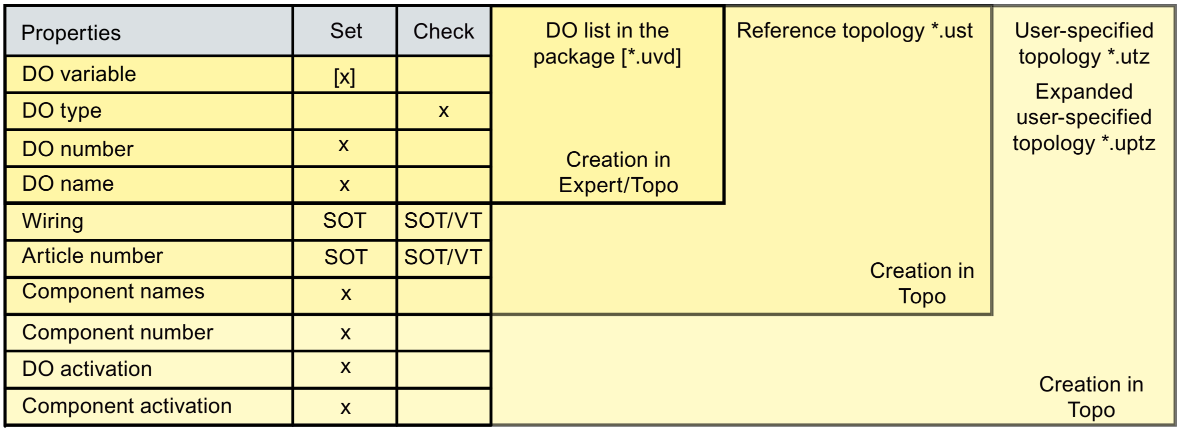
- The actual wiring of the SINAMICS in the control cabinet can be checked using the machine topology exported from SINAMICS. Warning or error messages are issued should deviations occur.
- You can check the properties of the components wired via DRIVE-CLiQ (DO type, article number, etc.). Wildcards are 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. When processing the package, the OEM component number in the DO1 is written to the 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 change of the DO name is configured, then this is automatically entered into the SINAMICS parameter p199 .
  DO number
  With the drive device configuration, SINAMICS assigns the DO numbers for all DOs at the CU and the NXs, i.e. on the particular CU starting at 1. For a machine series, it is more understandable, 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. The value range for the DO numbers lies between 2 and 54.
  Component names
  The component names can be freely assigned. 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. When the user-specified topology is used, the component numbers must also be specified. The CU and the NX must contain 1; the numbers on the CU or NX must not be assigned more than once and must be in the range 2 to 199. In Topo, the component numbers can also be automatically assigned via the shortcut menu.
- When configuring a user-specified topology, the activation status of the DOs and components can still be specified.
  The following is true:
  - 0 inactive and present
  - 1 active and present
  - 2 not present

  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 variable	[x]	
DO type		x
DO number	x	
DO name	x	
Wiring	SOT	SOT/VT
Article number	SOT	SOT/VT
Component names	x	
Component number	x	
DO activation	x	
Component activation	x	

DO list in the package [*.uvd]
Creation in Expert/Topo
Reference topology *.ust
Creation in Topo
User-specified topology *.utz
Expanded user-specified topology *.uptz
Creation in Topo

Figure 6-10 DO list - comparison topology (VT) - user-specified topology (SOT)

6.6.7 Configuration of the topology with Topo

CMC Topo can be used to configure not only a comparison topology but also a user-defined topology. The operation of CMC Topo is described in detail in Section Create MyConfig – Topo (Page 294).

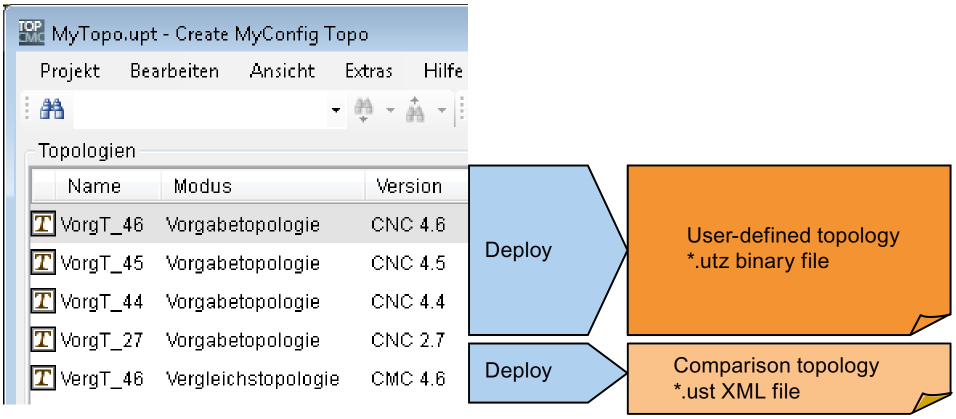
The wiring diagram for the DRIVE-CLiQ connections and some of the properties are identical for both topologies.

With the changeover from comparison topology to user-defined topology, the additional properties DO activation, component activation and component number must also be specified.

On switchover to comparison topology, the data remain stored for when switch-back to user-defined topology is performed. These data have no effect in the comparison topology.

For the deployment as user-defined topology, other necessary data is added internally and output as binary user-defined topology file.

To deploy a user-defined topology, it is necessary to specify a CNC software version.



Name	Modus	Version
VorgT_46	Vorgabetopologie	CNC 4.6
VorgT_45	Vorgabetopologie	CNC 4.5
VorgT_44	Vorgabetopologie	CNC 4.4
VorgT_27	Vorgabetopologie	CNC 2.7
VergT_46	Vergleichstopologie	CMC 4.6

Figure 6-11 Variants of the topology generation with CMC Topo

See also

Generation of the comparison topology using external software (Page 267)

6.6.8 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.

You will find more information on this in Section Interface - Topo-XML (Page 331).

This means that it is also possible to generate the comparison topology using external software, which is integrated in the PLM process of the machine manufacturer, e.g. ePlan, ELCAD or similar. When required, the externally generated comparison topologies can also be read-in in Topo as part of the production preparation for control purposes or adaptations, before they are linked into the project when configuring with Expert or placed next to the package as configuration file. A comparison configuration file, which is located in or next to the package, can also be selected manually on the dialog of the "SINAMICS topology" during the runtime. Normally, a special topology must be created for each machine version, whereas one package can be sufficient for all machines.

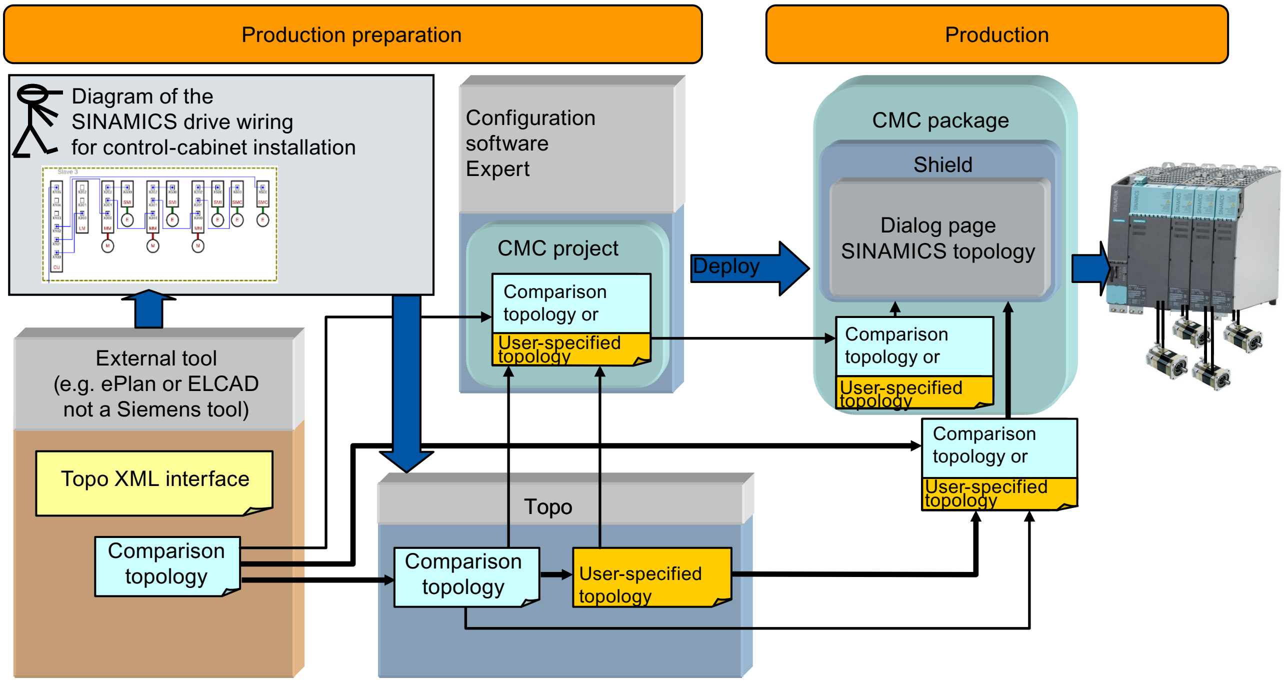


Figure 6-12 Configuration of a CMC package with comparison topology or user-specified topology

Compatibility of CMC versions and CNC software releases

The Topo XML interface is available in the version SCI 2.6, CMC V4.4/4.5 and CMC V4.6. For transfer, CMC Topo V4.6 no longer supports the Topo XML interfaces SCI 2.6 and CMC V4.4/4.5.

Note

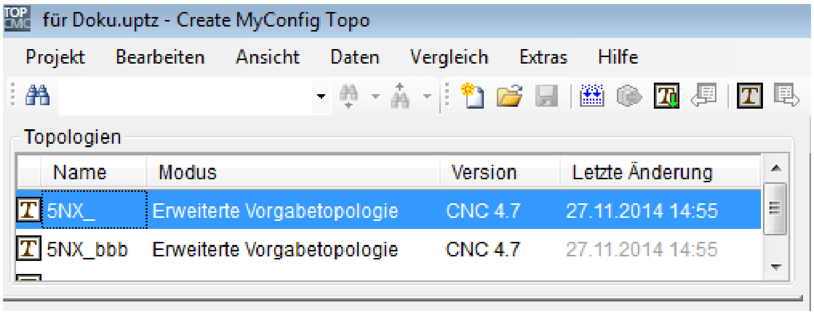
CMC Topo 4.6 can only generate comparison or user-specified topology for CMC packages 4.6.

For deployment of a user-defined topology, the CNC system software version must be defined to which this will be applied.
CMC Topo V4.6 supports the CNC software versions V2.6, V2.7, V4.4, V4.5, and V4.6.

6.7 Extended user-specified topology

Requirement

As of the Create MyConfig version V4.7 there is also the extended user-specified topology as a third mode in addition to the comparison and user-specified topology.



Name	Modus	Version	Letzte Änderung
5NX_	Erweiterte Vorgabetopologie	CNC 4.7	27.11.2014 14:55
5NX_bbb	Erweiterte Vorgabetopologie	CNC 4.7	27.11.2014 14:55

Figure 6-13 Extended user-specified topology

In addition to the SINAMICS wiring diagram, there is also NC and DRV data in the extended user-specified topology.

The starting basis for the extended user-specified topology is the archive import of a functioning machine.

After the import, the wiring diagram is displayed as for V4.6. In addition, additional NC and DRV data are available in the background for a manipulation.

All of the data remain available until it is deleted and can be saved as Topo project <Name>.uptz.

During the import, DO variables are automatically assigned to the drive objects (DO). These are needed for the drive or encoder assignment.

Components can be deleted, added or moved using the known mechanisms such as drag and drop, copy and paste.

Drive objects

The extended user-specified topology works only with complete drive objects.

This means that

- when a DO is moved, the complete dataset is always moved, including all of the components used in it.
- all of the data of the DO is symbolically displayed by the "Motor module" component (TM, hub).
- the complete DO is always considered when adding/deleting a motor module.

If individual measuring systems are processed, these must be changed via the data view (see below).

Note

Measuring system / Motor data

Note that it is not possible to assign a measuring system or motor data to a different DO by moving a component.

Example: moving a dual axis module from Slave 12 to Slave 13

In the following example, the selected dual motor module is moved from NX Slave 12 X103 to NX Slave 13 X102. During the move
a prompt appears, asking whether attached components, NC or drive data should also be moved or deleted.

The colored highlighting denotes the attached data.

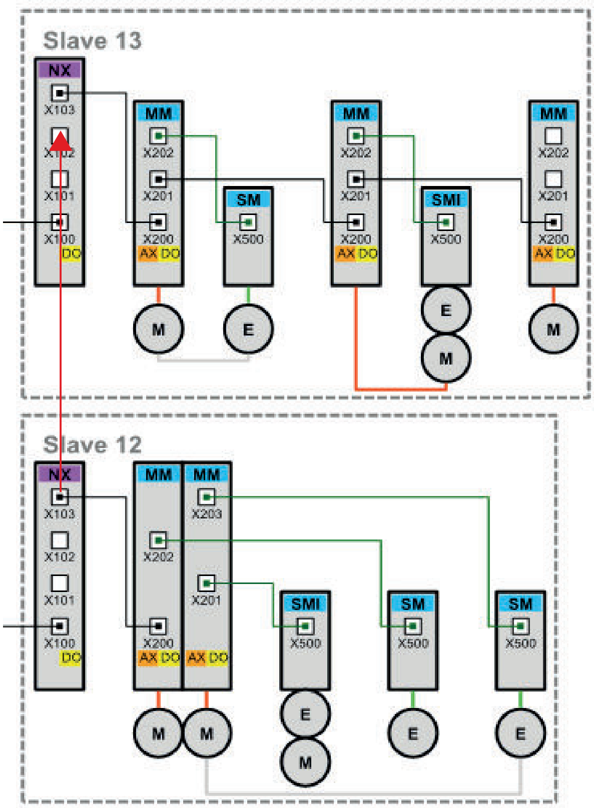


Figure 6-14 Moving a dual axis module from Slave 12 to Slave 13

Note

No more than 6 motor modules (MM) can be connected to one slave:

6.8 Axis-drive assignment

After the drives have been configured in SINAMICS, there is no assignment yet between the NC axes and servo drive objects. The desired assignment must to be explicitly written into a number of NC machine data.
Configuration is possible with CMC. After automatic device configuration or user-defined topology, a service drive object (DO) of a topology is automatically assigned to each NC axis. Configuration takes place in Topo. The necessary NC data are written during execution of the package.

if an axis-drive assignment has been performed on a machine, Topo can perform a diagnosis using the NC-DRV archive.

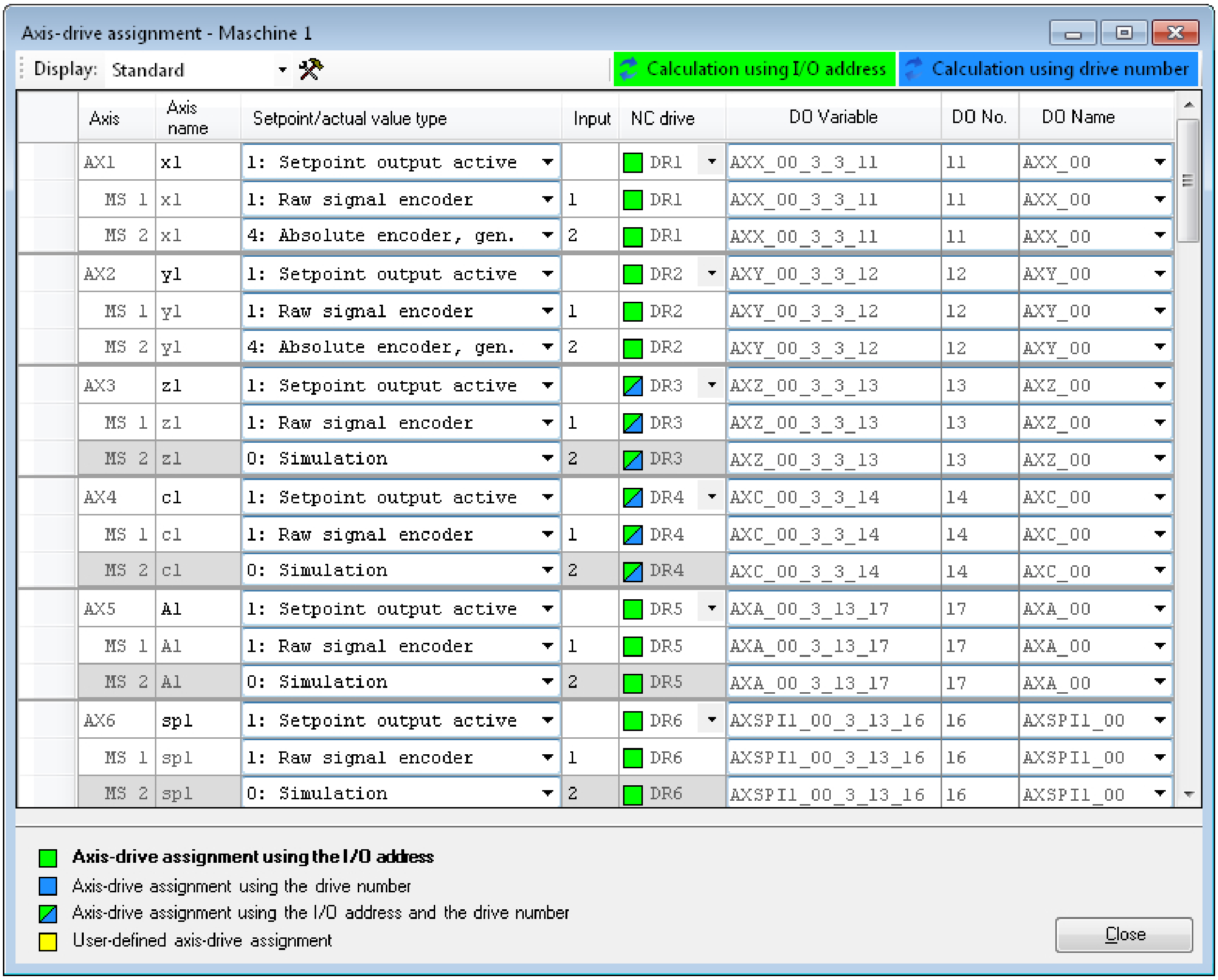
Configuration

First of all, the topology must be available as a comparison or user-defined topology and DO variables must be assigned to the component of the servo drive objects.

The table for configuring the axis-drive assignment is called using the **Tools** menu or the icon. Only under **Display: Standard** is configuration possible.

The data are displayed that are relevant for configuration. Assignment of the machine data, e.g. I/O address, will not be performed until during runtime of the package and cannot be displayed at the time of configuration.

Example of a tabular view for configuration



Axis-drive assignment - Maschine 1

Display: Standard

Calculation using I/O address | Calculation using drive number

Axis	Axis name	Setpoint/actual value type	Input	NC drive	DO Variable	DO No.	DO Name
AX1	x1	1: Setpoint output active		🟩 DR1	AXX_00_3_3_11	11	AXX_00
MS 1	x1	1: Raw signal encoder	1	🟩 DR1	AXX_00_3_3_11	11	AXX_00
MS 2	x1	4: Absolute encoder, gen.	2	🟩 DR1	AXX_00_3_3_11	11	AXX_00
AX2	y1	1: Setpoint output active		🟩 DR2	AXY_00_3_3_12	12	AXY_00
MS 1	y1	1: Raw signal encoder	1	🟩 DR2	AXY_00_3_3_12	12	AXY_00
MS 2	y1	4: Absolute encoder, gen.	2	🟩 DR2	AXY_00_3_3_12	12	AXY_00
AX3	z1	1: Setpoint output active		◪ DR3	AXZ_00_3_3_13	13	AXZ_00
MS 1	z1	1: Raw signal encoder	1	◪ DR3	AXZ_00_3_3_13	13	AXZ_00
MS 2	z1	0: Simulation	2	◪ DR3	AXZ_00_3_3_13	13	AXZ_00
AX4	c1	1: Setpoint output active		◪ DR4	AXC_00_3_3_14	14	AXC_00
MS 1	c1	1: Raw signal encoder	1	◪ DR4	AXC_00_3_3_14	14	AXC_00
MS 2	c1	0: Simulation	2	◪ DR4	AXC_00_3_3_14	14	AXC_00
AX5	A1	1: Setpoint output active		🟩 DR5	AXA_00_3_13_17	17	AXA_00
MS 1	A1	1: Raw signal encoder	1	🟩 DR5	AXA_00_3_13_17	17	AXA_00
MS 2	A1	0: Simulation	2	🟩 DR5	AXA_00_3_13_17	17	AXA_00
AX6	sp1	1: Setpoint output active		🟩 DR6	AXSPI1_00_3_13_16	16	AXSPI1_00
MS 1	sp1	1: Raw signal encoder	1	🟩 DR6	AXSPI1_00_3_13_16	16	AXSPI1_00
MS 2	sp1	0: Simulation	2	🟩 DR6	AXSPI1_00_3_13_16	16	AXSPI1_00

🟩 **Axis-drive assignment using the I/O address**
🟦 Axis-drive assignment using the drive number
◪ Axis-drive assignment using the I/O address and the drive number
🟨 User-defined axis-drive assignment

Figure 6-15 Table axis-drive assignment - representation standard

The table for configuration of the axis-drive assignment is composed of the following table columns:

Table column	Meaning
Axis	Display of the axes AX1 to AX31, including the two possible measurement systems MS1 and MS2.
Axis name	For configuration, the name of the axis that is written into the NC machine data `N10000 $MN_AXCONF_MACHAX_NAME_TAB` during package execution must be entered here.

Setpoint/actual value type	In this column, the Setpoint output can be activated via a list box and the encoder type can be specified. These data are entered in the following NC machine data:
	`N30130 $MA_CTRLOUT_TYPE`
	`N30240 $MA_ENC_TYPE`
	`N30200 $MA_NUM_ENCS`
	Activated axes and measurement systems are displayed on a white background.

Table column	Meaning
Input	In this column, the input can be modified for the measurement systems. This states whether it is the first or the second encoder.
NC Drive	The "NC drive" column is not activated by Topo until after assignment of the DO variable and contains the drive number. This assignment can be influenced if no automatic assignment is possible.
DO variable	The columns "DO variable," "DO No.," and "DO name" are coupled to each other to form a drive object. A DO variable must be assigned to the NC axis via a list box. All the drive objects are available that have been assigned to the motor modules or sensor modules in the topology. **Note** Selection is only possible if the DO variable table is correctly filled out.
DO No.	
DO name	

Assigning axes to the drive

There are two ways of assigning the axes to the drive:

1. Axis-drive assignment by adaptation of the I/O addresses -
   In this setting, the drive number is set to be equal to the axis number and the NC address machine data are adapted.
   ```
   N13050 $MN_DRIVE_LOGIC_ADDRESS
   N10393 $MN_SAFE_DRIVE_LOGIC_ADDRESS
   ```
2. Axis-drive assignment by adaptation of the drive number -
   With this setting, an attempt is made to adapt the drive number in he NC axis machine data in such a way that the I/O addresses remain at the default values.
   ```
   N30110 $MA_CTRLOUT_MODULE_NR..
   N30220 $MA_ENC_MODULE_NR
   N36906 $MA_SAFE_CTRLOUT_MODULE_NR
   N10000 $MN_AXCONF_MACHAX_NAME_TAB[?]
   ```

Note

In special cases, such as CU320, setpoint switchover etc., no automatic assignment can be made. In this case, it is up to the configurer to select the NC drive. This is the configurer's own responsibility. A manual selection is always marked with yellow in the "NC drive" column.
In the bottom status area/legend, the assigned colors are explained and the most important axis-drive assignment is highlighted.
The table corresponds to the topology representation, i.e. if an assignment has occurred and an axis has been clicked, the associated drive object will be selected in the topology image.

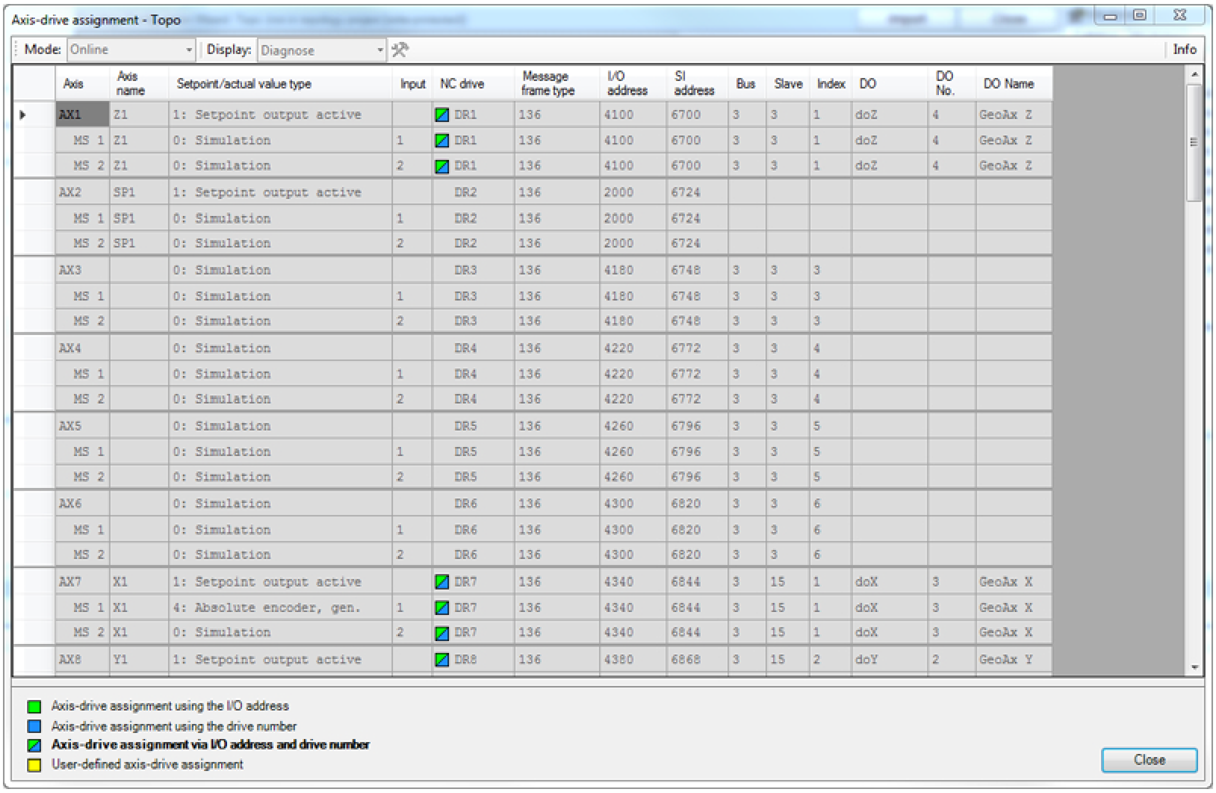
Diagnostics

If you import an NC/DRC archive via the Project > Display/import topology... menu, the topology will be shown on a gray background.
When you open the "Axis-drive assignment" dialog box, you see the "Display: Diagnosis". This contains all consistent information required to represent the exact assignment of the relevant machine data as they have been set on the machine by automatic axis-drive assignment or manually.

This table is used for diagnostics and checking the axis-drive assignment. If, with the "Import" button, you take over the topology on a gray background into the topology list of Topo, you can perform another configuration and the displayed list under "Diagnostics" becomes invalid.

When you open the "Axis-drive assignment" dialog, after importing in the "Axis-drive assignment" dialog, you will see the "Display: Standard". You can change the configuration. The axis drive dialog with "Display: Diagnostics" is not available for the topologies in the topology list.

Example of a tabular view for diagnostics



Axis-drive assignment - Topo

Mode: Online | Display: Diagnose

Axis	Axis name	Setpoint/actual value type	Input	NC drive	Message frame type	I/O address	SI address	Bus	Slave	Index	DO	DO No.	DO Name
AX1	Z1	1: Setpoint output active		◩ DR1	136	4100	6700	3	3	1	doZ	4	GeoAx Z
MS 1	Z1	0: Simulation	1	◩ DR1	136	4100	6700	3	3	1	doZ	4	GeoAx Z
MS 2	Z1	0: Simulation	2	◩ DR1	136	4100	6700	3	3	1	doZ	4	GeoAx Z
AX2	SP1	1: Setpoint output active		DR2	136	2000	6724						
MS 1	SP1	0: Simulation	1	DR2	136	2000	6724						
MS 2	SP1	0: Simulation	2	DR2	136	2000	6724						
AX3		0: Simulation		DR3	136	4180	6748	3	3	3			
MS 1		0: Simulation	1	DR3	136	4180	6748	3	3	3			
MS 2		0: Simulation	2	DR3	136	4180	6748	3	3	3			
AX4		0: Simulation		DR4	136	4220	6772	3	3	4			
MS 1		0: Simulation	1	DR4	136	4220	6772	3	3	4			
MS 2		0: Simulation	2	DR4	136	4220	6772	3	3	4			
AX5		0: Simulation		DR5	136	4260	6796	3	3	5			
MS 1		0: Simulation	1	DR5	136	4260	6796	3	3	5			
MS 2		0: Simulation	2	DR5	136	4260	6796	3	3	5			
AX6		0: Simulation		DR6	136	4300	6820	3	3	6			
MS 1		0: Simulation	1	DR6	136	4300	6820	3	3	6			
MS 2		0: Simulation	2	DR6	136	4300	6820	3	3	6			
AX7	X1	1: Setpoint output active		◩ DR7	136	4340	6844	3	15	1	doX	3	GeoAx X
MS 1	X1	4: Absolute encoder, gen.	1	◩ DR7	136	4340	6844	3	15	1	doX	3	GeoAx X
MS 2	X1	0: Simulation	2	◩ DR7	136	4340	6844	3	15	1	doX	3	GeoAx X
AX8	Y1	1: Setpoint output active		◩ DR8	136	4380	6868	3	15	2	doY	2	GeoAx Y

🟩 Axis-drive assignment using the I/O address
🟦 Axis-drive assignment using the drive number
◩ **Axis-drive assignment via I/O address and drive number**
🟨 User-defined axis-drive assignment

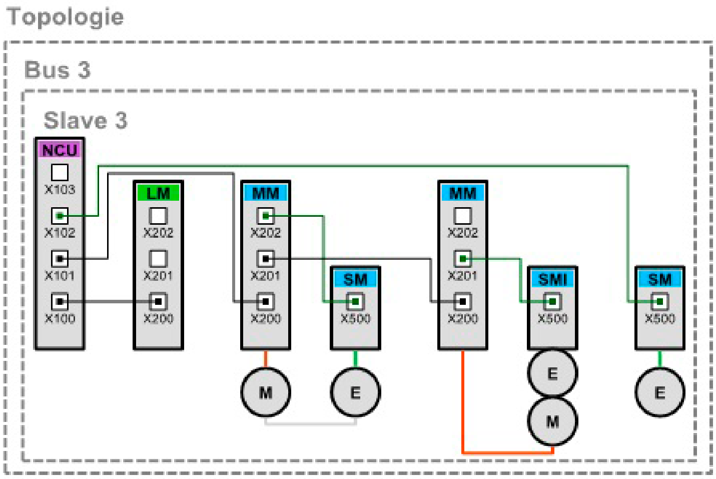
Figure 6-16 Table axis-drive assignment - representation diagnostics

Configuration Dialog

A configuration dialog box is opened with the ⚒ icon. In the dialog box, the "standard" and "diagnostics" variants offered in the display can be changed with regard to the columns shown and new user-defined display variants can be added.

6.9 Interface Topo XML

6.9.1 Example for displaying the wiring using an XML image



```
<?xml version="1.0" encoding="UTF-8"?>
<dp preliminary="true" build="008" appVersion="4.7.0.2"
version="4.7.0.0">
  <bus subsystem="3">
    <slave address="3">
      <control_unit_ncu doname="NCU_XY" donr="1" dotype="3"
dovar="doCU" name="NCU" compnr="1">
        <cliq index="0">
          <line_module doname="ALM_XY" donr="2" dotype="3003"
dovar="doLM" name="ALM" compnr="2">
            <cliq index="0" uplink="true"/>
            <cliq index="1"/>
            <cliq index="2"/>
          </line_module>
        </cliq>
        <cliq index="1">
          <motor_module doname="X-drive" donr="3" dotype="11"
dovar="doX" name="MM_X" compnr="3">
            <cliq index="0" uplink="true"/>
            <cliq index="1">
              <motor_module doname="Y-drive" donr="4" dotype="11"
dovar="doY" name="MM_Y" compnr="7">
                <cliq index="0" uplink="true"/>
                <cliq index="1">
                  <sensor_module name="SM1_Y" compnr="9" sm="1"
doref="doY" group="DQI_Y">
                    <cliq index="0" uplink="true"/>
                   <encoder name="Enc1_Y" compnr="10" group="DQI_Y"/
>
```

```
                    <motor name="Motor_Y" compnr="11" group="DQI_Y"/
>
                  </sensor_module>
                </cliq>
                <cliq index="2"/>
              </motor_module>
            </cliq>
            <cliq index="2">
            <sensor_module name="SM_X" compnr="5" sm="1" doref="doX">
                <cliq index="0" uplink="true"/>
                <encoder name="Encoder_X" compnr="6"/>
              </sensor_module>
            </cliq>
            <motor name="Motor_X" compnr="4"/>
          </motor_module>
        </cliq>
        <cliq index="2">
          <sensor_module name="SM2_Y" compnr="8" sm="2" doref="doY">
            <cliq index="0" uplink="true"/>
            <encoder name="Enc2_Y" compnr="12"/>
          </sensor_module>
        </cliq>
        <cliq index="3"/>
      </control_unit_ncu>
    </slave>
  </bus>
</dp>
```

Explanation

`<element>`	black, bold	Tag element
`attr=".."`	black, normal	Required attribute
`attr=".."`	gray, normal	An attribute that is only required in a certain constellation
`attr=".."`	blue, italics	Optional attributes for the parameterization of the package (variables)
`attr=".."`	brown, italics	Optional attributes to change the component name
`attr=".."`	green, italics	Optional attributes to restrict to certain components when checking

6.9.2 Description of the XML structure

SINAMICS components

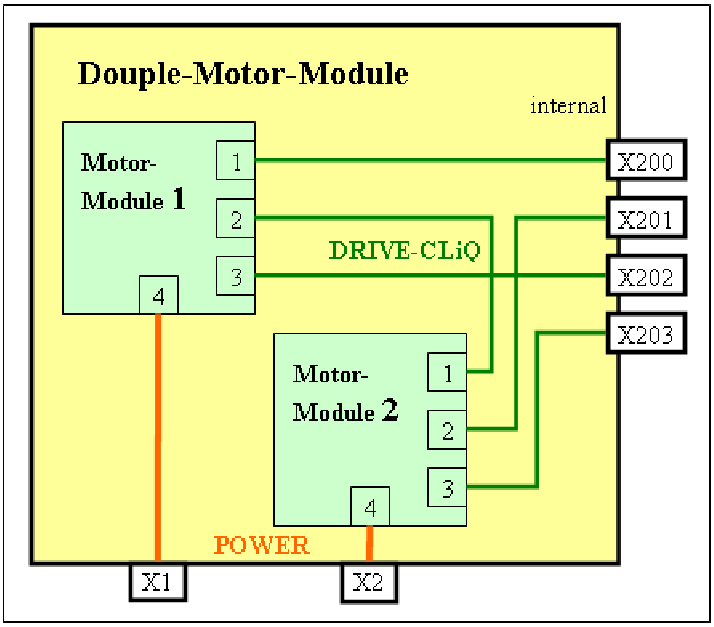
SINAMICS components should be specified in the structure as XML elements based on their device class. To further reduce the recognition tolerance level, restrictive attributes can be specified (order number).

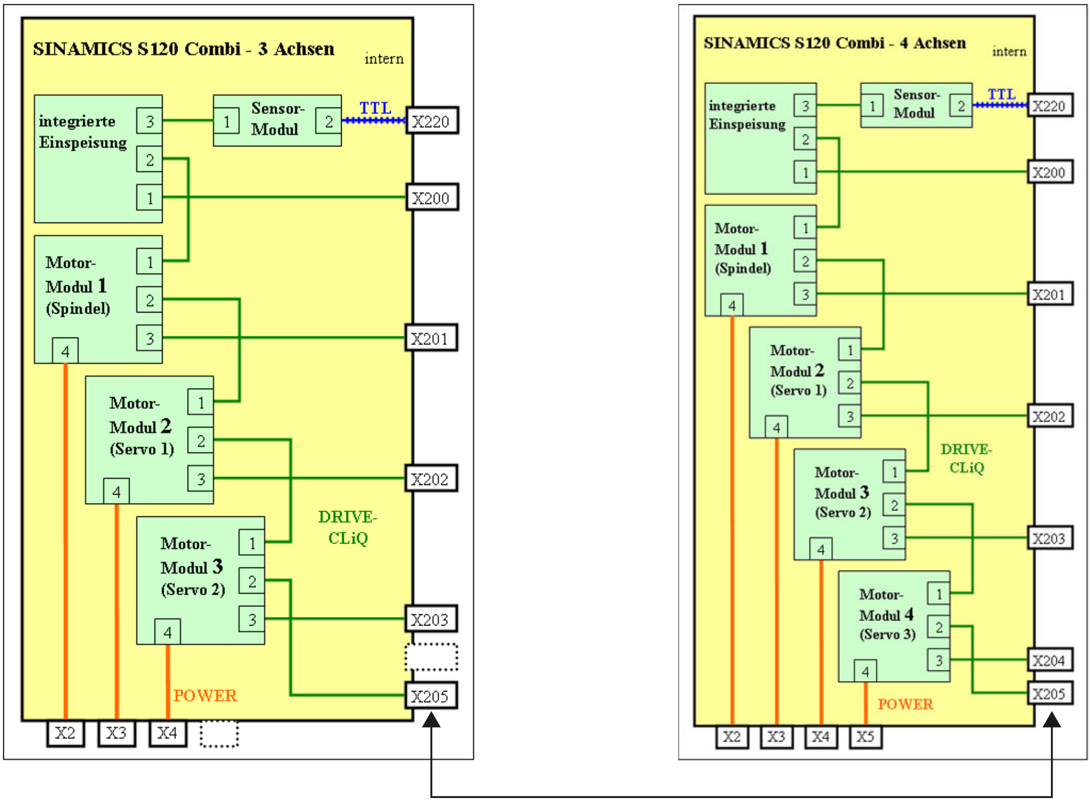
Device classes:

control_unit_ncu, control_unit_nx, control_unit_cu,
line_module, voltage_sensing_module,
motor_module, sensor_module, motor, encoder,
motor_module_hf, damping_module, sine_filter_module
hub_module, terminal_module, terminal_board
motor_module_hf, damping_module, sine_filter_module

Composite modules

For composite modules, such as Double Motor Modules and 120 combi modules, their internal structure must be mapped in the XML structure. To ensure the correct display as composite module in Topo, all the internal components belonging to such a module must be identified with the attribute group="<name of the composite component>". In addition, the "index" attribute with the position inside the module must be specified (MM1 → index=1, etc.) for components of the same device class (here Motor Module).





In both
versions
X205

□ DRIVE-CLiQ connection type

○ Analog connection type

▽ Power connection type

Figure 6-17 S120 Combi topological modeling

Automatically created DOs for DRIVE-CLiQ components

SINAMICS automatically creates DOs for specific DRIVE-CLiQ components.

Automatically created DOs for DRIVE-CLiQ components:

control_unit_ncu,
motor_module_hf control_unit_nx, control_unit_cu, line_module, motor_module, hub_module,
terminal_module,
terminal_board.

SINAMICS components for which a DO is created, can be assigned a "dovar" attribute. Using the "dovar" attribute, the component is assigned a DO variable. This DO variable contains the properties of the DO that can be used subsequently by the package when executing the package.
If such a component was assigned a DO variable manually or via the Topo XML file, this DO

variable can also be used to specify the user-specific DO name and DO number. For each component that is assigned a DO variable, an appropriate type must also be specified via the "dotype" attribute.

The possible DO types are:

- SINAMICS S
- Integrated (NCU)
- Extended (NX)
- ACTIVE INFEED CONTROL
- SERVO
- SMART INFEED CONTROL
- BASIC INFEED CONTROL
- TB30
- DMC20 / DME20
- TM31
- TM41
- TM15
- TM120
- CULINK
- Determine INFEED from hardware
- Determine TM from hardware

DOs not created automatically for DRIVE-CLiQ components

DRIVE-CLiQ components for which no DO was generated by SINAMICS can be assigned existing DOs.

DOs not created automatically for DRIVE-CLiQ components:

sensor_module,
damping_module voltage_sensing_module

This is realized by specifying "doref". "doref" references a DO and thus specifies the association of the component to the DO. To do this, the same DO variable is used as that specified for the required DO via the "dovar" attribute. This allows, for example, Sensor Modules on a hub to be assigned to a SERVO-DO.
The "sm" attribute of the "sensor_module" element designates the encoder as 1st or 2nd encoder of the DO. It must then be used if the encoder cannot be assigned uniquely.

For modeling DQI/SMI motors, a **motor** must be specified on the **sensor_module** along with the **encoder**. The motor must also be removed from the associated **motor_module**.
The associated components must be identified with the attribute **group**="<name of the composite component>".

```
<motor_module nam=“eMM1" dovar="var1" dotype="11" donr="20"
doname="do">
   <cliq index="0" uplink="true"/>
   <cliq index="1"/>
   <cliq index="2">
      <sensor_module name="SM1" group="DQI_01" doref="var1" sm="1">
             <cliq index="0" uplink="true"/>
                     <encoder name="Enc1" group="DQI_01"/>
                     <motor name="Mot1" group="DQI_01"/>
         </sensor_module>
   </cliq>
</motor_module>
```

Non-DRIVE-CLiQ components (motor, encoder, sine_filter_module) must be adequately defined using their subordinate hierarchical position.

The "cliq" element designates a port of a DRIVE-CLiQ component. The "index" attribute specifies the last position of the terminal designation (for X104 → 4, for X501 → 1, etc.).

The "uplink=true" attribute in the "CLiQ" element specifies that it involves an "incoming connection" from a higher-level component.

Each component can be assigned an ID using the "oemid" attribute.
This ID is written to SINAMICS parameter p7859[component number] in DO1. Using this parameter, the machine manufacturer can address additional information, such as additional error texts, spare part numbers, etc.

In order to use a comparison topology to generate a user-specified topology without further manual processing in Topo, a unique name as well as a unique component number for each slave ("compnr/compnr2" attribute) must be assigned for each component.

To deactivate individual components of a DO or whole DOs, the "compstate" and "dostate" attributes with appropriate valid values can be assigned to the corresponding components.

6.9.3 XSD for XML-interface

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation>
      Content:        Topo XML interface
      Version: 4.7
      Date: 01.09.2014
    </xs:documentation>
  </xs:annotation>

  <xs:simpleType name="uInt">
    <xs:union>
      <xs:simpleType>
        <xs:restriction base="xs:unsignedInt"/>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value=""/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>

  <xs:simpleType name="uShort">
    <xs:union>
      <xs:simpleType>
        <xs:restriction base="xs:unsignedShort"/>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value=""/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>

  <xs:simpleType name ="smnr">
    <xs:restriction base="xs:unsignedShort">
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="state">
    <xs:restriction base="xs:unsignedShort">
      <xs:enumeration value="0"/>
      <!--inactive-->
      <xs:enumeration value="1"/>
      <!--active-->
      <xs:enumeration value="2"/>
      <!--inactive / not available-->
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="dotype">
    <xs:restriction base="xs:unsignedInt">
      <xs:enumeration value="1"/> <!--CU (CU320/CU310)-->
      <xs:enumeration value="3"/> <!--NCU-->
      <xs:enumeration value="4"/> <!--NX-->

      <xs:enumeration value="10"/> <!--ALM-->
      <xs:enumeration value="20"/> <!--SLM-->
      <xs:enumeration value="30"/> <!--BLM-->
      <xs:enumeration value="3003"/> <!--Determine DOType LM from HW
(only SOT as of 4.6)-->

      <xs:enumeration value="11"/> <!--SERVO-->
      <xs:enumeration value="70"/><!--SERVO HLA-->

      <xs:enumeration value="200"/> <!--TM31-->
      <xs:enumeration value="201"/> <!--TM41-->
      <xs:enumeration value="204"/> <!--TM15-->
      <xs:enumeration value="207"/> <!--TM120-->
      <xs:enumeration value="3006"/> <!--Determoine DOType TM from
HW (only SOT as of 4.6)-->

      <xs:enumeration value="100"/> <!--TB30-->

      <xs:enumeration value="150"/> <!--HUB-->

      <xs:enumeration value="254"/> <!--CULINK-->
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name ="adamode">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ONLINE"/>
      <xs:enumeration value="NONE"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name ="axdrnr">
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="31"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name ="encidx">
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name ="ctrltype">
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="4"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name ="enctype">
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="5"/>
    </xs:restriction>
  </xs:simpleType>
  <!--
  <xs:simpleType name ="slotnr">
    <xs:restriction base="xs:unsignedInt">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="6"/>
    </xs:restriction>
  </xs:simpleType>
  -->
  <xs:element name="dp">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="bus" type="bus" minOccurs="1"
maxOccurs="unbounded"/>
        <xs:element name="ada" type="ada" minOccurs="0"
maxOccurs="1" />
      </xs:sequence>
      <xs:attribute name="version" type="xs:string" use="required"
fixed="4.7.0.0"/>
      <xs:attribute name="utzversion" type="xs:string"
use="optional" />
      <xs:attribute name="mode" type="xs:string" use="optional" />

      <!--only CMC!!! (not permitted for externals)-->
      <xs:attribute name="appVersion" type="xs:string"
use="optional" />
      <xs:attribute name="sciVersion" type="xs:string"
use="optional" />
      <xs:attribute name="build" type="xs:string" use="optional" />
      <xs:attribute name="preliminary" type="xs:boolean"
use="optional" />
      <!--only CMC!!! (not permitted for externals)-->
    </xs:complexType>
  </xs:element>

  <xs:complexType name="bus">
    <xs:sequence>
      <xs:element name="slave" type="slave" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="subsystem" type="xs:unsignedInt"
use="required" />
  </xs:complexType>

  <xs:complexType name="slave">
    <xs:choice minOccurs="1" maxOccurs="1">
      <xs:element name="control_unit_ncu" type="control_unit_ncu"
minOccurs="1" maxOccurs="1" />
      <xs:element name="control_unit_cu" type="control_unit_cu"
minOccurs="1" maxOccurs="1" />
      <xs:element name="control_unit_nx" type="control_unit_nx"
minOccurs="1" maxOccurs="1" />
```

```
    </xs:choice>
    <xs:attribute name="address" type="xs:unsignedInt"
use="required"/>
  </xs:complexType>

  <xs:complexType name="cliq">
    <xs:choice minOccurs="0" maxOccurs="1">
     <xs:element name="line_module" type="line_module" minOccurs="0"
maxOccurs="1" />
      <xs:element name="motor_module" type="motor_module"
minOccurs="0" maxOccurs="1" />
      <xs:element name="sensor_module" type="sensor_module"
minOccurs="0" maxOccurs="1" />
      <xs:element name="voltage_sensing_module"
type="voltage_sensing_module" minOccurs="0" maxOccurs="1" />
      <xs:element name="cu_link" type="cu_link" minOccurs="0"
maxOccurs="1" />
      <xs:element name="hub_module" type="hub_module" minOccurs="0"
maxOccurs="1" />
      <xs:element name="terminal_module" type="terminal_module"
minOccurs="0" maxOccurs="1" />
      <xs:element name="motor_module_hf" type="motor_module_hf"
minOccurs="0" maxOccurs="1" />
      <xs:element name="damping_module" type="damping_module"
minOccurs="0" maxOccurs="1" />
      <xs:element name="motor_module_hla" type="motor_module_hla"
minOccurs="0" maxOccurs="1" />
    </xs:choice>
    <xs:attribute name="index" type="xs:unsignedInt" use="required" /
>
    <xs:attribute name="uplink" type="xs:boolean" use="optional"
default="false" />
  </xs:complexType>

  <xs:complexType name="compbase" abstract="true">
    <xs:attribute name="name" type="xs:string" use="optional" />
    <xs:attribute name="compnr" type="uShort" use="optional" />
    <xs:attribute name="mlfb" type="xs:string" use="optional" />
    <xs:attribute name="oemid" type="uInt" use="optional" />
    <xs:attribute name="rds" type="xs:string" use="optional" />
    <xs:attribute name="group" type="xs:string" use="optional" />
    <xs:attribute name="index" type="xs:unsignedInt" use="optional" /
>
  </xs:complexType>

  <xs:complexType name="docomp" abstract="true">
    <xs:complexContent>
      <xs:extension base="compbase">
        <xs:attribute name="dovar" type="xs:string" use="optional" /
>
        <xs:attribute name="dotype" type="dotype" use="optional" />
        <xs:attribute name="donr" type="uShort" use="optional" />
        <xs:attribute name="doname" type="xs:string" use="optional" /
>
        <xs:attribute name="dodesc" type="xs:string" use="optional" /
>
      </xs:extension>
   </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="dostatecomp" abstract="true">
    <xs:complexContent>
      <xs:extension base="docomp">
        <xs:attribute name="dostate" type="state" use="optional" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="dorefcomp" abstract="true">
    <xs:complexContent>
      <xs:extension base="compbase">
        <xs:attribute name="doref" type="xs:string" use="optional" /
>
        <xs:attribute name="compstate" type="state" use="optional" /
>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="control_unit_ncu">
    <xs:complexContent>
      <xs:extension base="docomp">
        <xs:sequence>
          <xs:element name="cliq" type="cliq" minOccurs="4"
maxOccurs="6" />
          <xs:choice minOccurs="1" maxOccurs="1">
            <xs:element name="terminal_board" type="terminal_board"
minOccurs="0" maxOccurs="1" />
            <xs:element name="comm_board" type="comm_board"
minOccurs="0" maxOccurs="1" />
          </xs:choice>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="control_unit_ncu_828">
    <xs:complexContent>
      <xs:extension base="docomp">
        <xs:sequence>
          <xs:element name="cliq" type="cliq" minOccurs="3"
maxOccurs="3" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="control_unit_cu">
    <xs:complexContent>
      <xs:extension base="docomp">
        <xs:sequence>
          <xs:element name="cliq" type="cliq" minOccurs="1"
maxOccurs="4" />
          <xs:choice minOccurs="1" maxOccurs="1">
            <xs:element name="terminal_board" type="terminal_board"
minOccurs="0" maxOccurs="1" />
            <xs:element name="comm_board" type="comm_board"
minOccurs="0" maxOccurs="1" />
          </xs:choice>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="control_unit_nx">
    <xs:complexContent>
      <xs:extension base="docomp">
        <xs:sequence>
          <xs:element name="cliq" type="cliq" minOccurs="4"
maxOccurs="4" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="line_module">
    <xs:complexContent> <xs:extension base="dostatecomp">
     <xs:sequence> <xs:element name="cliq" type="cliq" minOccurs="3"
maxOccurs="3" />
        </xs:sequence>
        <xs:attribute name="compstate" type="state" use="optional" /
>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="motor_module_base" abstract ="true">
    <xs:complexContent>
      <xs:extension base="dostatecomp">
        <xs:sequence>
          <xs:element name="cliq" type="cliq" minOccurs="3"
maxOccurs="3" />
        </xs:sequence>
        <xs:attribute name="compstate" type="state" use="optional" /
>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="motor_module">
    <xs:complexContent>
      <xs:extension base="motor_module_base">
        <xs:sequence>
          <xs:element name="motor" type="motor" minOccurs="0"
maxOccurs="1" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="motor_module_hf">
    <xs:complexContent>
      <xs:extension base="motor_module_base">
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="damping_module">
    <xs:complexContent>
      <xs:extension base="dorefcomp">
        <xs:choice maxOccurs="2">
          <xs:element name="cliq" type="cliq" minOccurs="3"
maxOccurs="3" />
          <xs:element name="sine_filter_module"
type="sine_filter_module" minOccurs="0" maxOccurs="1" />
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="motor_module_hla">
    <xs:complexContent>
      <xs:extension base="dostatecomp">
        <xs:choice maxOccurs="2">
          <xs:element name="cliq" type="cliq" minOccurs="4"
maxOccurs="4" />
          <xs:element name="valve" type="valve" minOccurs="0"
maxOccurs="1" />
        </xs:choice> <xs:attribute name="compstate" type="state"
use="optional" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="sensor_module">
    <xs:complexContent>
      <xs:extension base="dorefcomp">
        <xs:sequence>
```

```
          <xs:element name="cliq" type="cliq" minOccurs="1"
maxOccurs="1" />
          <xs:element name="encoder" type="encoder" minOccurs="0"
maxOccurs="1" />
          <!--only for SMI-->
          <xs:element name="motor" type="motor" minOccurs="0"
maxOccurs="1" />
          <!--only for SMI-->
        </xs:sequence>
        <xs:attribute name="sm" type="smnr" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="voltage_sensing_module">
    <xs:complexContent>
      <xs:extension base="dorefcomp">
       <xs:sequence>
         <xs:element name="cliq" type="cliq" minOccurs="1"
maxOccurs="1" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="sine_filter_module">
    <xs:complexContent>
      <xs:extension base="compbase">
        <xs:sequence>
          <xs:element name="motor" type="motor" minOccurs="0"
maxOccurs="1" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="motor">
    <xs:complexContent>
      <xs:extension base="compbase"/>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="encoder">
    <xs:complexContent>
      <xs:extension base="compbase"/>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="valve">
    <!--Valve for HLAModule-->
    <xs:complexContent>
      <xs:extension base="compbase">
```

```
        <xs:sequence>
          <xs:element name="actuator" type="actuator" minOccurs="0"
maxOccurs="1" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="actuator">
    <!--Actuator for HLAModule-->
    <xs:complexContent>
      <xs:extension base="compbase"/>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="hub_module">
    <xs:complexContent>
      <xs:extension base="dostatecomp">
        <xs:sequence>
          <xs:element name="cliq" type="cliq" minOccurs="6"
maxOccurs="6" />
        </xs:sequence>
        <xs:attribute name="compnr2" type="uShort" use="optional" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="terminal_module">
    <xs:complexContent>
      <xs:extension base="dostatecomp">
        <xs:sequence>
          <xs:element name="cliq" type="cliq" minOccurs="2"
maxOccurs="2" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="terminal_board">
    <xs:complexContent>
      <xs:extension base="dostatecomp"/>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="comm_board">
  </xs:complexType>

  <xs:complexType name="cu_link">
    <xs:complexContent>
      <xs:extension base="dostatecomp">
        <xs:attribute name="address" type="xs:unsignedInt"
use="optional" />
```

```
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ada">
    <xs:sequence>
      <xs:element name="ada_ctrl" type="ada_ctrl" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="mode" type="adamode" />
  </xs:complexType>

  <xs:complexType name="ada_ctrl">
    <xs:sequence>
      <xs:element name="ada_enc" type="ada_enc" minOccurs="0"
maxOccurs="2"/>
    </xs:sequence>
    <xs:attribute name="axnr" type="axdrnr" use="required" />
    <xs:attribute name="axname" type="xs:string" use="optional" />
    <xs:attribute name="drnr" type="axdrnr" use="optional"/>
    <xs:attribute name="ctrltype" type="ctrltype" use="optional"/>
    <xs:attribute name="dovar" type="xs:string" use="optional"/>
  </xs:complexType>

  <xs:complexType name="ada_enc">
    <xs:attribute name="idx" type="encidx" use ="required"/>
    <xs:attribute name="enctype" type="enctype" use="optional"/>
    <xs:attribute name="encinputnr" type="smnr" use="optional"/>
    <xs:attribute name="doref" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:schema>
```

Create MyConfig - NcuShare 7

7.1 Introduction

7.1.1 NcuShareService

An exchange release is required for processing Windows packages with NCU share. The previous configuration of the exchange release is replaced with the NcuShareService "Create MyConfig NcuShare".

A setup is available for the installation of the service.

7.2 Installation

7.2.1 Installation

There are three possible options for installing the service "Create MyConfig NcuShare".

1. Installation with the setup of Create MyConfig
2. Manual installation by executing NcuShareServiceSetup.msi
   After installing Create MyConfig, the setup file is available in the installation folder under / Shield.
3. Installation for package execution. (Reference to Expert)

When the service is installed, a folder can be configured for the installation types 1 and 2, which is used as the
exchange folder.

7.2.2 "Create MyConfig NcuShare" service

The service is installed during the installation on the Windows computer under the program name "Create MyConfig NcuShareService" with the specified service version and entered under the name "Create MyConfig NcuShare" in the list of services.
The service is in the service status "Finished" after installation.

The service is started with the execution of Windows packages with NCU share for the duration of the package execution and is then closed again.

When the service is started, it automatically activates the exchange release for the NCU access.
This permits the exchange of data between NCU and PC/PCU during package execution.

7.3 Error checking

What must be checked in the event of an error?

Use of virtual machines (VMWare)

In connection with virtual machines (VMWare), problems may occur when establishing a connection, because the release of the NCU cannot be achieved.

To ensure the NcuShareService functions properly, the virtual machine must be operated in "Bridged Mode". In "Bridged Mode", the release can be achieved by another computer while the service is running.

Utilization of the service on Windows XP (PCU)

To use the NcuShareService in Windows XP, it is necessary to deactivate the "simple file release" in order to allow authentication of the NCU.

The following link has a description of how you can deactivate a simple file release: http://support.microsoft.com/kb/307874 (http://support.microsoft.com/kb/307874)

Created user

During installation, the user "UpNcuUser" is created and the password is automatically assigned.

For security reasons, the password is changed each time the "UpNcuShareService" is started.

Note

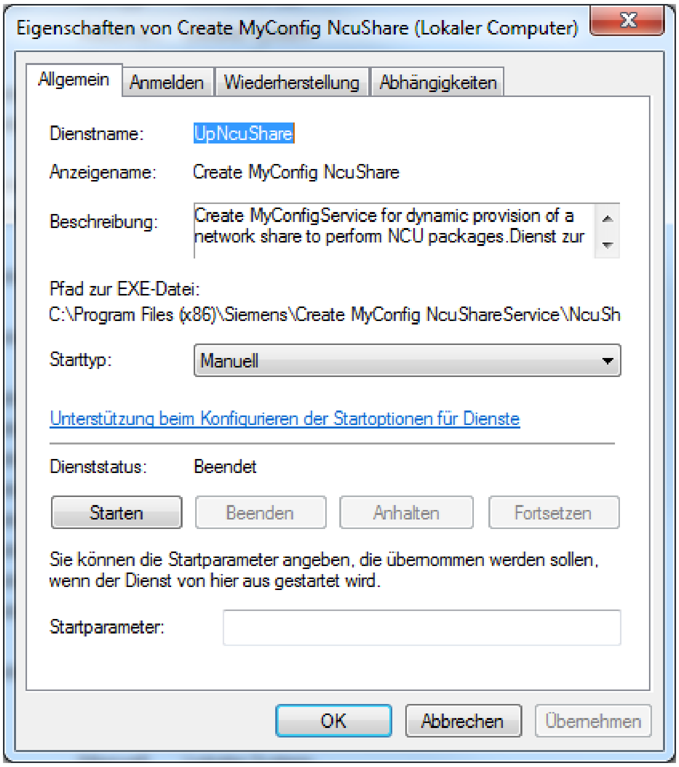
If no user is available when the "UpNcuShareService" is started, a new one is automatically created.

Configuration of the service

The service automatically ends and deletes the release if no shield is carried out.

Observe the following when setting the properties of the Create MyConfig NcuShare:

1. In the General tab, select the Type of start "Manually".



Eigenschaften von Create MyConfig NcuShare (Lokaler Computer)

Dienstname: UpNcuShare

Anzeigename: Create MyConfig NcuShare

Beschreibung: Create MyConfigService for dynamic provision of a network share to perform NCU packages.Dienst zur

Pfad zur EXE-Datei:
C:\Program Files (x86)\Siemens\Create MyConfig NcuShareService\NcuSh

Starttyp: Manuell

Unterstützung beim Konfigurieren der Startoptionen für Dienste

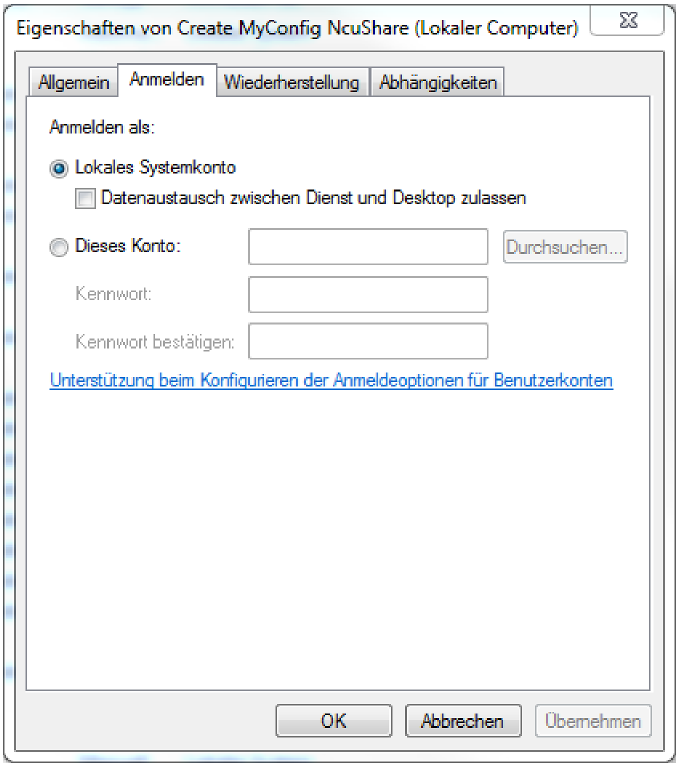
Dienststatus: Beendet

Sie können die Startparameter angeben, die übernommen werden sollen, wenn der Dienst von hier aus gestartet wird.

Startparameter:

Figure 7-1 Create MyConfig NcuShare - General tab

2. In the Login tab, select the option "Local system account".



Eigenschaften von Create MyConfig NcuShare (Lokaler Computer)

Anmelden als:

● Lokales Systemkonto

□ Datenaustausch zwischen Dienst und Desktop zulassen

○ Dieses Konto:

Kennwort:

Kennwort bestätigen:

Figure 7-2 Create MyConfig NcuShare - Login tab

Changing the release folder

The release folder is defined during the installation of the "UpNcuShareService".

In case of an installation with Create MyConfig or manual setups, the user can change the path during the installation.

If the "UpNcuShareService" is installed during the package execution, the default path is always used.

To subsequently change the exchange folder, the "UpNcuShareService" must first be uninstalled and then re-installed.

List of abbreviations

A

A.1 Abbreviations

AAZ	Axis-drive assignment
AGK	Automatic device configuration (per HMI)
CMC	Create MyConfig
CNC	Computerized Numerical Control
CU	Control Unit
DB	Data Block
DO	Drive Object
DRV	Drive Data
EUNA	End User Notification Administration; is replaced as from Version V4.7 by the Service Platform Industry (GSP).
FB	Function Block
FC	Function
GSP	Global Service Platform; is officially designated as Service Platform Industry (GSP)
HW Config	Hardware Configuration (Step 7)
NC	Numerical Control
NCK	Numerical Control Kernel
NCU	Numerical Control Unit (example: NCU 720.2)
OB	Organization Block
PC	Personal Computer
PCU	Personal Computer Unit (display device; example: PCU 50: Windows-based display device)
PG	Programming device
PLC	Programmable Logic Controller
SDB	System Data Block
SOT	SINAMICS offline topology package
TCU	Thin Client Unit
VT	Comparison topology

Table A-1 File extensions for the file formats used in CMC

*.arc	NC, PLC, SDB, or DRV archive
*.tgz	
*.ucz	Expert component
*.upt	Topology project; designation up to CNC Version V4.6
*.uptz	Topology project; the designation applies **as from** CMC Version **V4.7**
*.upz	Expert project

*.uss	Step XML, step properties
*.ust	Comparison topology
*.usz	Executable Linux package
*.utz	User-specified topology

Index