SIEMENS

# SINUMERIK

# SINUMERIK 840D sl Basesoftware and operating software

Commissioning Manual


Preface

Fundamental safety instructions 1

SINUMERIK Operate (IM9) 2

Easy XML 3

SINUMERIK Integrate Run MyScreens (BE2) 4

NCU operating system 5

PCU base software (IM8) 6

PCU base software (IM10) 7

Appendix A


Valid for
Control:
SINUMERIK 840D sl/840DE sl
Software:
CNC software V4.8 SP3
SINUMERIK Operate V4.8 SP3


08/2018
6FC5397-1DP40-6BA2

## 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

Document order number: 6FC5397-1DP40-6BA2
℗ 08/2018 Subject to change

Copyright © Siemens AG 2006 - 2018.
All rights reserved

# Preface

## SINUMERIK documentation

The SINUMERIK documentation is organized into the following categories:

- General documentation/catalogs
- User documentation
- Manufacturer/service documentation

## Additional information

You can find information on the following topics at the following address (https://support.industry.siemens.com/cs/de/en/view/108464614):

- Ordering documentation/overview of documentation
- Additional links to download documents
- Using documentation online (find and search in manuals/information)

If you have any questions regarding the technical documentation (e.g. suggestions, corrections), please send an e-mail to the following address (mailto:docu.motioncontrol@siemens.com).

## mySupport/Documentation

At the following address (https://support.industry.siemens.com/My/ww/en/documentation), you can find information on how to create your own individual documentation based on Siemens' content, and adapt it for your own machine documentation.

## Training

At the following address (http://www.siemens.com/sitrain), you can find information about SITRAIN (Siemens training on products, systems and solutions for automation and drives).

## FAQs

You can find Frequently Asked Questions in the Service&Support pages under Product Support (https://support.industry.siemens.com/cs/de/en/ps/faq).

## SINUMERIK

You can find information about SINUMERIK at the following address (http://www.siemens.com/sinumerik).

## Target group

This documentation is intended for commissioning personnel.

The plant or system is readily assembled and wired. For the following steps, e.g. configuring the individual components, the Commissioning Manual contains all necessary information or at least references.

## Benefits

The intended target group can use the Commissioning Manual to test and commission the system or the plant correctly and safely.

Utilization phase: Setup and commissioning phase

## Standard scope

This documentation only describes the functionality of the standard version. Additions or revisions made by the machine manufacturer are documented by the machine manufacturer.

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

For the sake of simplicity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation, or maintenance.

## Technical Support

Country-specific telephone numbers for technical support are provided in the Internet at the following address (https://support.industry.siemens.com/sc/ww/en/sc/2090) in the "Contact" area.

## CompactFlash Cards

- Do not remove the memory card while it is being accessed. This can lead to damage of the memory card and the SINUMERIK as well as the data on the memory card.
- Insert the memory card carefully and the right way round into the memory card slot (observe indicators such as arrow or similar). This way you avoid mechanical damage to the memory card or the device.
- Only use memory cards that have been approved by Siemens for use with SINUMERIK. Even though SINUMERIK follows general industry standards for memory cards, it is possible that memory cards from some manufacturers will not function perfectly in this device or are not completely compatible with it (you can obtain information on compatibility from the memory card manufacturer or supplier).
- For further information on handling CompactFlash cards, please refer to the NCU manuals.

# Table of contents


**Preface** ..... **3**

**1 Fundamental safety instructions** ..... **7**

1.1 General safety instructions ..... 7

1.2 Equipment damage due to electric fields or electrostatic discharge ..... 11

1.3 Warranty and liability for application examples ..... 12

1.4 Industrial security ..... 13

1.5 Residual risks of power drive systems ..... 15

**2 SINUMERIK Operate (IM9)** ..... **17**

**3 Easy XML** ..... **567**

**4 SINUMERIK Integrate Run MyScreens (BE2)** ..... **791**

**5 NCU operating system** ..... **1137**

**6 PCU base software (IM8)** ..... **1239**

**7 PCU base software (IM10)** ..... **1363**

**A Appendix** ..... **1537**

A.1 Documentation overview SINUMERIK 840D sl ..... 1537

# Fundamental safety instructions

# 1

## 1.1 General safety instructions



**⚠ WARNING**

**Electric shock and danger to life due to other energy sources**

Touching live components can result in death or severe injury.

- Only work on electrical devices when you are qualified for this job.
- Always observe the country-specific safety rules.

Generally, the following six steps apply when establishing safety:

1. Prepare for disconnection. Notify all those who will be affected by the procedure.
2. Isolate the drive system from the power supply and take measures to prevent it being switched back on again.
3. Wait until the discharge time specified on the warning labels has elapsed.
4. Check that there is no voltage between any of the power connections, and between any of the power connections and the protective conductor connection.
5. Check whether the existing auxiliary supply circuits are de-energized.
6. Ensure that the motors cannot move.
7. Identify all other dangerous energy sources, e.g. compressed air, hydraulic systems, or water. Switch the energy sources to a safe state.
8. Check that the correct drive system is completely locked.

After you have completed the work, restore the operational readiness in the inverse sequence.



**⚠ WARNING**

**Electric shock due to connection to an unsuitable power supply**

When equipment is connected to an unsuitable power supply, exposed components may carry a hazardous voltage that might result in serious injury or death.

- Only use power supplies that provide SELV (Safety Extra Low Voltage) or PELV-(Protective Extra Low Voltage) output voltages for all connections and terminals of the electronics modules.



**⚠ WARNING**

**Electric shock due to equipment damage**

Improper handling may cause damage to equipment. For damaged devices, hazardous voltages can be present at the enclosure or at exposed components; if touched, this can result in death or severe injury.

- Ensure compliance with the limit values specified in the technical data during transport, storage and operation.
- Do not use any damaged devices.



**⚠ WARNING**

**Electric shock due to unconnected cable shields**

Hazardous touch voltages can occur through capacitive cross-coupling due to unconnected cable shields.

- As a minimum, connect cable shields and the cores of cables that are not used at one end at the grounded housing potential.



**⚠ WARNING**

**Electric shock if there is no ground connection**

For missing or incorrectly implemented protective conductor connection for devices with protection class I, high voltages can be present at open, exposed parts, which when touched, can result in death or severe injury.

- Ground the device in compliance with the applicable regulations.

**⚠ WARNING**

**Spread of fire from built-in devices**

In the event of fire outbreak, the enclosures of built-in devices cannot prevent the escape of fire and smoke. This can result in serious personal injury or property damage.

- Install built-in units in a suitable metal cabinet in such a way that personnel are protected against fire and smoke, or take other appropriate measures to protect personnel.
- Ensure that smoke can only escape via controlled and monitored paths.

**⚠ WARNING**

**Unexpected movement of machines caused by radio devices or mobile phones**

When radio devices or mobile phones with a transmission power > 1 W are used in the immediate vicinity of components, they may cause the equipment to malfunction. Malfunctions may impair the functional safety of machines and can therefore put people in danger or lead to property damage.

- If you come closer than around 2 m to such components, switch off any radios or mobile phones.
- Use the "SIEMENS Industry Online Support app" only on equipment that has already been switched off.

**⚠ WARNING**

**Fire due to inadequate ventilation clearances**

Inadequate ventilation clearances can cause overheating of components with subsequent fire and smoke. This can cause severe injury or even death. This can also result in increased downtime and reduced service lives for devices/systems.

- Ensure compliance with the specified minimum clearance as ventilation clearance for the respective component.

**⚠ WARNING**

**Unexpected movement of machines caused by inactive safety functions**

Inactive or non-adapted safety functions can trigger unexpected machine movements that may result in serious injury or death.

- Observe the information in the appropriate product documentation before commissioning.
- Carry out a safety inspection for functions relevant to safety on the entire system, including all safety-related components.
- Ensure that the safety functions used in your drives and automation tasks are adjusted and activated through appropriate parameterizing.
- Perform a function test.
- Only put your plant into live operation once you have guaranteed that the functions relevant to safety are running correctly.

**Note**

**Important safety notices for Safety Integrated functions**

If you want to use Safety Integrated functions, you must observe the safety notices in the Safety Integrated manuals.

**⚠ WARNING**

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

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

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

## 1.2 Equipment damage due to electric fields or electrostatic discharge

Electrostatic sensitive devices (ESD) are individual components, integrated circuits, modules or devices that may be damaged by either electric fields or electrostatic discharge.



**NOTICE**

**Equipment damage due to electric fields or electrostatic discharge**

Electric fields or electrostatic discharge can cause malfunctions through damaged individual components, integrated circuits, modules or devices.

- Only pack, store, transport and send electronic components, modules or devices in their original packaging or in other suitable materials, e.g conductive foam rubber of aluminum foil.
- Only touch components, modules and devices when you are grounded by one of the following methods:
  - Wearing an ESD wrist strap
  - Wearing ESD shoes or ESD grounding straps in ESD areas with conductive flooring
- Only place electronic components, modules or devices on conductive surfaces (table with ESD surface, conductive ESD foam, ESD packaging, ESD transport container).

## 1.3 Warranty and liability for application examples

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

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

## 1.4 Industrial security

**Note**

**Industrial security**

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

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

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

For additional information on industrial security measures that may be implemented, please visit:

Industrial security (http://www.siemens.com/industrialsecurity)

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

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at:

Industrial security (http://www.siemens.com/industrialsecurity)

Further information is provided on the Internet:

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

**⚠ WARNING**

**Unsafe operating states resulting from software manipulation**

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

- Keep the software up to date.
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
- Make sure that you include all installed products into the holistic industrial security concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.
- Protect the drive against unauthorized changes by activating the "know-how protection" drive function.

## 1.5 Residual risks of power drive systems

When assessing the machine- or system-related risk in accordance with the respective local regulations (e.g., EC Machinery Directive), the machine manufacturer or system installer must take into account the following residual risks emanating from the control and drive components of a drive system:

1. Unintentional movements of driven machine or system components during commissioning, operation, maintenance, and repairs caused by, for example,
   - Hardware and/or software errors in the sensors, control system, actuators, and cables and connections
   - Response times of the control system and of the drive
   - Operation and/or environmental conditions outside the specification
   - Condensation/conductive contamination
   - Parameterization, programming, cabling, and installation errors
   - Use of wireless devices/mobile phones in the immediate vicinity of electronic components
   - External influences/damage
   - X-ray, ionizing radiation and cosmic radiation
2. Unusually high temperatures, including open flames, as well as emissions of light, noise, particles, gases, etc., can occur inside and outside the components under fault conditions caused by, for example:
   - Component failure
   - Software errors
   - Operation and/or environmental conditions outside the specification
   - External influences/damage
3. Hazardous shock voltages caused by, for example:
   - Component failure
   - Influence during electrostatic charging
   - Induction of voltages in moving motors
   - Operation and/or environmental conditions outside the specification
   - Condensation/conductive contamination
   - External influences/damage
4. Electrical, magnetic and electromagnetic fields generated in operation that can pose a risk to people with a pacemaker, implants or metal replacement joints, etc., if they are too close
5. Release of environmental pollutants or emissions as a result of improper operation of the system and/or failure to dispose of components safely and correctly
6. Influence of network-connected communication systems, e.g. ripple-control transmitters or data communication via the network

For more information about the residual risks of the drive system components, see the relevant sections in the technical user documentation.

SIEMENS

# SINUMERIK

# SINUMERIK 840D sl
# SINUMERIK Operate (IM9)

Commissioning Manual


System overview 1
Principles of start-up 2
General settings 3
Licensing 4
Creating drive connections 5
Channel menu 6
Machine and setting data 7
Customizing the "Machine" operating area 8
Simulation and simultaneous recording 9
Spindle functions 10
Drive system 11
Configuring alarms 12
Data backup 13
Configuring the network 14
Service and diagnostics 15
OEM-specific online help 16
SINUMERIK Integrate - AMB, AMC, AMM 17
Technologies and cycles 18
Additional language 19
SINUMERIK Operate on PCU/IPC 20
HT 8 21
List of abbreviations/ acronyms A


Valid for
Control:
SINUMERIK 840D sl/840DE sl
Software:
CNC software V4.8 SP3
SINUMERIK Operate V4.8 SP3


08/2018
6FC5397-1DP40-6BA2

## 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

Document order number: 6FC5397-1DP40-6BA2
℗ 08/2018 Subject to change

Copyright © Siemens AG 2006 - 2018.
All rights reserved

# Table of contents


**1 System overview....13**

1.1 System overview....13

1.2 System requirements....14

**2 Principles of start-up....17**

2.1 Data structure of SINUMERIK Operate....17

2.2 Editing files....20
2.2.1 Editing the file at the controller....20
2.2.2 Editing a file externally....22

**3 General settings....25**

3.1 Changing the language....25

3.2 Set date/time....26

3.3 Using the time server....28

3.4 Use time zone....29

3.5 Configuring Caps Lock....30

3.6 Activating/deactivating the virtual keyboard....31

3.7 Defining the access rights for the "HMI restart" softkey....31

3.8 Changing the skin design....32

3.9 Setting the screensaver....32

3.10 Inserting a user-specific run up screen....33

3.11 Using user interfaces in parallel....33

3.12 Setting how login data (credentials) are handled when archiving....34

3.13 VNC remote connections....35

3.14 Access levels....39
3.14.1 Definition of access levels....39
3.14.2 Modifying the access levels password....40
3.14.3 Access levels for programs....41
3.14.4 Access rights for files....44
3.14.5 Access levels for softkeys....44
3.14.6 This is how you define new access levels for softkeys....46
3.14.7 Setting the MDA program access via protection level....47

3.15 Program selection....48
3.15.1 Displaying the softkey....48

3.16 Activate and configure the Sidescreen window....50
3.16.1 Activate Sidescreen....50
3.16.2 Display Sidescreen ABC keyboard....51

3.16.3 Display Sidescreen machine control panel....52
3.16.3.1 Configuring your own keys on the machine control panel....53
3.16.3.2 Create own translations of texts for the machine control panel....55
3.17 SINUMERIK Operate Display Manager....56
3.17.1 Display configurations....56
3.17.2 Creating your own display configurations....57
3.17.2.1 Displays....57
3.17.2.2 Frames....58
3.17.2.3 Menus....60
3.17.2.4 Menu actions....66
3.17.2.5 Applications....68
3.17.2.6 Global settings....73
**4 Licensing....75**
4.1 Determining the license requirement....76
4.2 Activating the test license....78
4.3 Exporting license requirements....79
4.4 Reading in the license key....79
4.5 Backing up the license of a new CompactFlash card....80
**5 Creating drive connections....83**
5.1 Setting up drives....83
5.2 Setting EXTCALL....84
5.3 Creating a global part program memory....85
**6 Channel menu....87**
6.1 Configuring a channel menu....87
6.2 Constraints....88
6.3 General structure of the channel menu....88
6.4 Defining channel groups in the "netnames.ini" file....89
6.5 Configuring operating stations in the "config.ini" file....90
6.6 Example: Settings for 1:N....91
6.7 Distribution via job lists in the case of 1:N....93
**7 Machine and setting data....97**
7.1 Machine and setting data....97
7.2 Displaying/editing machine data....97
7.3 Displaying/editing display machine data....100
7.4 Displaying/editing setting data....100
7.5 Displaying/editing drive parameters....102
7.6 Machine data display options....104
7.7 Editing machine data and drive parameters....106
7.7.1 Editing hexadecimal values....106

7.7.2 Editing BICO values....107
7.7.3 Editing enum values....107
7.8 Searching for data....108
7.9 Managing data....110
7.10 User views....112
7.10.1 Creating a user view....113
7.10.2 Editing the user view....114
7.10.3 Deleting a user view....116
7.11 Plain texts for machine and setting data....116
**8 Customizing the "Machine" operating area....119**
8.1 Setting the font size of the actual value display....119
8.2 Inserting a user-specific logo....120
8.3 Configuring the display of the G-code groups....121
8.4 Configuring the channel operational message....123
8.5 Deactivating program test....125
8.6 Display STAT and TU....125
8.7 Activating the "Teach In" function....126
8.8 Block search....128
8.8.1 Activating block search mode....128
8.8.2 Accelerated block search for execution from external....129
8.9 Multi-channel support....129
8.10 Manual machine....131
8.11 User status display (OEM)....133
8.12 Configuring the offset display....136
8.13 Activate machining time recording....137
**9 Simulation and simultaneous recording....139**
9.1 Simulation overview....139
9.2 Setting the technology for simulation....141
9.3 Simultaneous recording overview....143
9.4 Clamping a blank....144
**10 Spindle functions....149**
10.1 Spindle control....149
10.2 Analog spindle....150
10.3 Spindle diagnostics....151
10.3.1 Spindle diagnostics....151
10.3.2 Temperatures....153
10.3.3 Motor temperature sensor....155
10.3.4 Additional temperature sensor....155
10.3.5 Temperature histograms....156

10.3.6 Speed/torque....156
10.3.7 Clamping system....157
10.3.8 Clamping system: Speed limits....159
10.3.9 Clamping system: Diagnostic statistics....160
10.3.10 Clamping system: Clamping time statistics....161
10.3.11 Fetching the logistics data....161

**11 Drive system....163**

11.1 Commissioning of drives....163

**12 Configuring alarms....165**

12.1 Creating alarm and message texts via the user interface....165

12.2 Configuring alarm and message texts via alarm text files....167
12.2.1 Creating in-house alarm texts....168
12.2.2 Creating texts for indexed alarm parameters....170
12.2.3 Creating part program message texts....173
12.2.4 Changing alarm attributes....176
12.2.5 Replacing standard alarm texts....180
12.2.6 Range of alarms....184
12.2.7 Parameter specifications in alarm texts....185
12.2.8 Opening error file....186

12.3 Configuring an alarm log....187
12.3.1 Setting alarm logs from the user interface....188
12.3.2 Loading alarm log via configuration file....188

12.4 PLC alarms with parameters....192
12.4.1 Definition of a parameter of the octet string data type....193
12.4.2 Definition of the language-dependent formatting....195

12.5 Deactivating a warning....196

**13 Data backup....199**

13.1 Overview....199

13.2 Creating a start-up archive....200

13.3 Reading-in a start-up archive....203

13.4 Backing up the hardware configuration....204

13.5 Creating an archive with original data....205

13.6 Reading in an archive with original data....207

13.7 Generating the complete archive....208

13.8 Serial interface (V24 / RS232)....208
13.8.1 Reading-in and reading-out archives via a serial interface....208
13.8.2 Setting interface parameters....211

13.9 Backing up setup data....212

13.10 Network settings....213

**14 Configuring the network....217**

14.1 Displaying the network overview....217

14.2 Settings of the system network....218
14.3 Settings of the factory network....221
14.4 Saving network settings....223
14.5 Configuring operator panels....224
14.5.1 Displaying the available operator panels....224
14.5.2 Configuring VNC connections....225
14.5.3 T:M:N configuration....226
14.6 Station-related network diagnostics....227
14.6.1 Displaying network adapters....228
14.6.2 Error analysis....229
14.6.2.1 Performing error diagnostics....229
14.6.2.2 MCP/EKS as network participant cannot be accessed....230
14.6.2.3 HMI on the PCU cannot establish a network connection to the NC....231
14.6.2.4 TCU cannot establish a network connection to the HMI....235
14.6.2.5 The TCU cannot establish a network connection to the assigned PCU....237
14.6.3 Configuring TCP/IP diagnostics....238
14.6.4 Ethernet network diagnostics....242
14.6.4.1 Diagnostics of the network and DHCP server....242
14.6.4.2 Messages....243
14.6.4.3 Network trace....243
14.6.4.4 Accessible nodes....247
**15 Service and diagnostics....251**
15.1 NC/PLC variables....251
15.1.1 Displaying and editing PLC and NC variables....251
15.1.2 Saving and loading screen forms....255
15.2 Displaying the Service overview....256
15.2.1 Selecting axes and drives....257
15.2.2 Axis diagnostics....259
15.2.3 Service drive....264
15.3 System utilization....271
15.4 Creating screenshots....272
15.5 Machine identity....273
15.5.1 Entering machine-specific information....274
15.5.2 Create a template....275
15.5.3 Import template....277
15.5.4 Save information....279
15.5.5 Adding hardware components....280
15.5.6 Configuration data....282
15.6 Logbook....282
15.6.1 1. Documenting startup....283
15.6.2 2. Defining start-up....283
15.6.3 Making a logbook entry....284
15.7 Action log....285
15.7.1 Setting the action log....285
15.7.2 Displaying the log file....287
15.7.3 Searching in the log files....288

15.7.4 Storing a log....288
15.7.5 Structure of a log file....289
15.7.6 Advanced settings....291
15.8 HMI trace....292
15.9 PROFIBUS diagnosis....293
15.9.1 Displaying details for DP slaves....295
15.10 Drive system....296
15.10.1 Displaying drive states....296
15.10.2 Displaying details of the drive objects....297
15.11 Remote diagnostics....299
15.11.1 Adapting remote diagnostics....299
15.11.2 Remote diagnostics via Teleservice adapter IE at X127....300
15.11.3 PLC control for remote access....302
15.12 Trace....303
15.12.1 General procedure....304
15.12.2 Trace session....305
15.12.2.1 Creating a session file....305
15.12.2.2 Saving the trace file....305
15.12.2.3 Load trace session file....306
15.12.3 Variable for trace....307
15.12.3.1 Variables filter/search....307
15.12.3.2 Selecting attributes of a variable....309
15.12.3.3 Displaying details of a variable....312
15.12.4 Trace settings....313
15.12.4.1 Trace settings (PLC, NC, servo)....313
15.12.4.2 Trace settings (drive)....313
15.12.4.3 Trace options....315
15.12.4.4 Starting the trace....315
15.12.5 Evaluate a trace....316
15.12.5.1 Setting trace views....316
15.12.5.2 Selecting a variable....317
15.12.5.3 Scaling the display....318
15.12.5.4 Zooming the display....319
15.12.5.5 Position the cursor....320
15.12.5.6 Acquiring measured values....321
15.13 PROFIBUS/PROFINET and AS-i bus diagnostics....322
15.13.1 PROFIBUS/PROFINET....322
15.13.2 Displaying PROFIBUS/PROFINET diagnostics....323
15.13.3 Configuring the AS-i bus....324
15.13.4 Displaying AS-i bus diagnostics....326
**16 OEM-specific online help....329**
16.1 Overview....329
16.2 Generating HTML files....330
16.3 Generating the help book....333
16.4 Integrating the online help in SINUMERIK Operate....335
16.5 Saving help files....337

16.6 Generating online help for user alarms and machine data....337
16.7 Example: This is how you create an online help for NC/PLC variables....341
16.8 Example: This is how you create a programming online help....343
16.9 Help files in PDF format....345
**17 SINUMERIK Integrate - AMB, AMC, AMM....347**
**18 Technologies and cycles....349**
18.1 Activating turning/milling/drilling/grinding technologies....349
18.2 Technology cycles for drilling....354
18.3 Manufacturer cycles....356
18.3.1 Manufacturer cycles....356
18.3.2 Prerequisites for the simulation mode....357
18.3.3 Extending the PROG_EVENT standard cycle....358
18.3.4 Manufacturer cycle for tool change CUST_T and CUST_M6....359
18.3.5 CUST_TECHCYC.SPF manufacturer cycle....360
18.3.6 CUST_MULTICHAN user cycle....363
18.4 Milling....363
18.4.1 General....363
18.4.2 Technology cycles for milling....364
18.4.3 Setting-up ShopMill cycles for milling....365
18.4.4 Cylinder surface transformation (TRACYL)....368
18.4.5 Example: Milling machine with the XYZ-AC axis configuration....369
18.4.6 ShopMill cycles for multiple clamping....373
18.5 Turning....375
18.5.1 General....375
18.5.2 Setting up cycles for turning....375
18.5.3 Setting-up ShopTurn cycles for turning....379
18.5.4 Setting up a counterspindle....381
18.5.5 Setting up the counterspindle under ShopTurn....384
18.5.6 Technology cycles for turning....385
18.5.7 Axis configuration of a lathe....389
18.5.8 Cylinder surface transformation (TRACYL)....390
18.5.9 End face machining (TRANSMIT)....394
18.5.10 Inclined Y axis (TRAANG)....397
18.5.11 Spindle speed limitation for current program....400
18.6 Grinding....401
18.7 Swiveling....404
18.7.1 Technology cycles for swiveling....404
18.7.2 CYCLE800 checklist for the identification of the machine kinematics....413
18.7.3 Commissioning swivel data (machine kinematics)....414
18.7.4 Examples of machine kinematics for the commissioning of the Swivel function....425
18.7.5 Manufacturer cycle CUST_800.SPF....435
18.7.6 Indirectly programming the name of the swivel data set....441
18.7.7 CYCLE996 measuring workpiece kinematics....442
18.8 Turning on milling machines....444
18.8.1 Setting up machine data and setting data....444

18.8.2 Setting up a kinematics transformation....449
18.8.3 Behavior at reset and power on....451
18.8.4 Setting the tool clamping angle....452
18.8.5 Setting up the Hirth joint....453
18.8.6 Different coordinate systems for milling and turning....454
18.8.7 Adaptations for CUST_800....454
18.8.8 Rotary axis positioning during block search....457
18.8.9 Follow-up for rotary axis positioning after block search....457

18.9 High-speed machining....458
18.9.1 High speed settings: Configuring CYCLE832....458
18.9.2 Manufacturer cycle CUST_832.SPF....460
18.9.3 Top Surface: CYCLE832 extension....462

18.10 Measuring cycles and measurement functions....463
18.10.1 Measuring cycles and measurement functions, general....463
18.10.2 Manufacturer and user cycle CUST_MEACYC.SPF....466
18.10.3 Measuring in JOG mode....469
18.10.3.1 Measuring workpieces at the milling machines....472
18.10.3.2 Measuring tools at the milling machines....474
18.10.3.3 Measuring tools at the turning machines....479
18.10.3.4 Measuring without electronic probe in JOG....480
18.10.4 Measuring in the AUTOMATIC mode....480
18.10.4.1 Measuring workpieces, general....482
18.10.4.2 Measuring workpieces at the milling machines....485
18.10.4.3 Measuring tools at the milling machines....486
18.10.4.4 Measuring workpieces at the turning machines....492
18.10.4.5 Measuring tools at the turning machines....493
18.10.5 Logging....494
18.10.5.1 Logging the measuring, general....494
18.10.5.2 Logging while measuring in the JOG mode....495
18.10.5.3 Logging while measuring in the AUTOMATIC mode....495
18.10.5.4 Manufacturer and user cycle CUST_MEAPROT.SPF....496

18.11 Compare cycles version....497
18.11.1 Display cycles version....497
18.11.2 Specify cycles version....499

**19 Additional language....501**

19.1 Installing additional languages....501

19.2 Uninstalling languages....502

19.3 Supported languages....502

**20 SINUMERIK Operate on PCU/IPC....505**

20.1 Link OEMFrame application....505

20.2 Parameterizing the OEMFrame application....513

20.3 FindWindow program application....521

20.4 Activating/deactivating window mode....522

20.5 Setting text legibility by means of font smoothing....523

20.6 Using interactive or silent installation versions....524

20.7 Setting the IP address of the NCU....526
20.8 SINUMERIK Operate exiting....527
**21 HT 8....529**
21.1 Configuring the traversing keys....529
21.2 Configuring user-specific key labeling....533
21.3 Configuring the function display at user-specific keys (U keys)....536
**A List of abbreviations/acronyms....539**
**Index....543**

# System overview

# 1

## 1.1 System overview

**Note**

**PCU / IPC**

In this document, any textual use of PCU also applies to IPC.

### NCU

The SINUMERIK Operate V4.x operating software is an integral component of the CNC software V4.x and can therefore run on an NCU 7x0.3. The NCU's CompactFlash card is the standard data carrier of the CNC software.

The SINUMERIK Operate operating software is started automatically when the NCU software boots and in the delivery state offers its standard functional scope, which depends on the actual NC and PLC configuration.

### PCU/IPC

The SINUMERIK Operate operating software is supplied on DVD and must then be installed on the hard disk of the PCU/IPC. The operating software is automatically started when the controller boots.

### PC

The SINUMERIK Operate operating software is supplied on DVD and can be installed in a freely selectable directory.

The following sections describe the expansions and adaptations that the machine manufacturer can make using the SINUMERIK Operate operating software.

### References

For a description of the commissioning procedure for the SINUMERIK 840D sl controller, please refer to the following documentation:

CNC Commissioning Manual: NCK, PLC, Drive

The integration of SINUMERIK Operate in WinCC / TIAP is described in the following documentation:

SINUMERIK Integrate Create MyHMI /WinCC Configuration Manual

# 1.2 System requirements

## Overview

It is a requirement that all system components be fully mounted, installed and configured. Commissioning of the NC and PLC must have been completed. The PLC is in the "Run" condition.

- Storage medium, e.g. USB FlashDrive and/or network connection: Handles data exchange with an external PC/PG.

**Note**

**USB keyboard**

You can use a USB keyboard to enter uppercase and lowercase text.

**Note**

When you load user data with a USB FlashDrive, the data size must not be larger than 4 GB!

## SINUMERIK Operate under Windows

You require the following software if you use SINUMERIK Operate under Windows:

- PCU 50.5: WinXP PCU basic software as of V1.2
- PCU 50.5: Win7 PCU basic software as of V11.4
- IPC: Win7 PCU basic software as of V11.4
- PC: Windows XP SP3 / Windows 7 / Windows 10 operating system

## Additional programs

- TextPad text editor: Used for the external creation and editing of XML and INI files in Windows. You can get the program on the Internet at:
  http://www.textpad.com
- WinSCP: Enables protected data transfer between different computers (NCU - PC). You can get the program on the Internet at:
  http://winscp.net/eng/index.php
- SINUMERIK Integrate Access MyMachine /P2P (previously RCS Commander): Permits the remote maintenance for machines with the SINUMERIK Operate operating software from a standard Windows PC.
- SINUMERIK Integrate Create MyConfig: Due to the modular concept, it enables automatic series production and the upgrade of different machines of a series with one package.
- Access MyMachine provides a special conversion function to transfer alarm texts of HMI Advanced or HMI Embedded to SINUMERIK Operate.

## References

A detailed description of the structure and commissioning of the system network with controller and operator components is provided in the following description:

SINUMERIK 840D sl/840Di sl Operator Components and Networking

# Principles of start-up

# 2

## 2.1 Data structure of SINUMERIK Operate

For SINUMERIK Operate, all files are saved on the CompactFlash card (NCU) or to the hard disk (PCU/IPC) or DVD (PC).

The file system is at the uppermost level:

- On the NCU in the "/System CF-Card/" directory
- On the PCU/IPC in the directory "C:\Program Files (x86)\Siemens\MotionControl"
- On the PC, the drive can be freely selected, e.g. "C:\Programs\siemens\sinumerik

The directory structure comprises the following directories:

- addon
- compat (PC/PCU/IPC only)
- oem
- siemens
- user

These directories have essentially an identical structure.

**Note**

The files in the “siemens” directory represent the original status and cannot be modified!

Files, whose contents can be supplemented or modified, are available as templates (specimen files) in the following directories:

| Files | Directory |
|---|---|
| Configuration files | /siemens/sinumerik/hmi/template/cfg |
| Text files | /siemens/sinumerik/hmi/template/lng |

Before making any changes/supplements, these files should be copied into the /cfg or /lng in /**oem**/sinumerik/hmi/, /**addon**/sinumerik/hmi/ or /**user**/sinumerik/hmi/ directories.

### Structure

The section of the directory structure relevant for SINUMERIK Operate is shown below:

| siemens directory | | | |
|---|---|---|---|
| /siemens/sinumerik | | | |
| | /hmi | | |
| | | /appl | // applications (operating areas) |
| | | /base | // basis system components |

| siemens directory | | | |
|---|---|---|---|
| | /cfg | | // all configuration files |
| | /data | | // version data |
| | /hlp | | // online help files |
| | /hlps | | // online help files, zipped and version files |
| | /ico | | // symbol files |
| | | /Ico640 | // icons in resolution 640x480 |
| | | /ico800 | // icons in resolution 800x600 |
| | | /ico1024 | // icons in resolution 1024x768 |
| | | /ico1280 | // icons in resolution 1280x1024 |
| | | /ico1600 | // icons in resolution 1600x1240 |
| | /lng | | // text files |
| | /lngs | | // text files zipped and versions files |
| | /osal | | |
| | | /ace | // ACE/TAO |
| | | /qt | // Qt |
| | /proj | | // EasyScreen configuring |
| | /template | | // various templates |
| | | /cfg | // templates for configuration files |
| | | /ing | // templates for text files |
| | /tmpp | | // storage, temporary data |
| /sys_cache/hmi | | | // various log files |

| compat directory | | |
|---|---|---|
| /compat | | // files for integrating the OEMFrame applications |
| | /add_on | |
| | /hmi_adv | |
| | /mmc2 | |
| | /oem | |
| | /user | |

| addon directory | | | |
|---|---|---|---|
| /addon/sinumerik | | | |
| | /hmi | | |
| | /appl | | // applications (operating areas) |
| | /cfg | | // configuration files |
| | /data | | // version data |
| | /hlp | | // online help files, zipped and version files |
| | /ico | | // symbol files |
| | | /Ico640 | // icons in resolution 640x480 |
| | | /ico800 | // icons in resolution 800x600 |
| | | /ico1024 | // icons in resolution 1024x768 |

| addon directory | | | |
|---|---|---|---|
| | | /ico1280 | // icons in resolution 1280x1024 |
| | | /ico1600 | // icons in resolution 1600x1240 |
| | /lng | | // text files |
| | /lngs | | // text files zipped and versions files |
| | /proj | | // EasyScreen configuring |
| | /template | | // various templates |

| oem directory | | | |
|---|---|---|---|
| /oem/sinumerik | | | |
| | /data | | // version data |
| | | /archive | // manufacturer archive |
| | /hmi | | |
| | | /appl | // applications (operating areas) |
| | | /cfg | // configuration files |
| | | /data | // version data |
| | | /hlp | // online help files |
| | | /hlps | // online help files, zipped and version files |
| | | /ico | // symbol files |
| | | | /Ico640 // icons in resolution 640x480 |
| | | | /ico800 // icons in resolution 800x600 |
| | | | /ico1024 // icons in resolution 1024x768 |
| | | | /ico1280 // icons in resolution 1280x1024 |
| | | | /ico1600 // icons in resolution 1600x1240 |
| | | /lng | // text files |
| | | /lngs | // text files zipped and versions files |
| | | /proj | // EasyScreen configuring |
| | | /template | // various templates |

| user directory | | | |
|---|---|---|---|
| /user/sinumerik | | | |
| | /data | | // version data |
| | | /archive | // user-specific archive |
| | | /prog | // user-specific programs |
| | /hmi | | |
| | | /cfg | // configuration files |
| | | /data | // version data |
| | | /hlp | // online help files |
| | | /ico | // symbol files |
| | | | /Ico640 // icons in resolution 640x480 |
| | | | /ico800 // icons in resolution 800x600 |
| | | | /ico1024 // icons in resolution 1024x768 |

| user directory | | |
|---|---|---|
| | /ico1280 | // icons in resolution 1280x1024 |
| | /ico1600 | // icons in resolution 1600x1240 |
| /lng | | // text files |
| /proj | | // EasyScreen configuring |
| /log | | // log files |
| /md | | // machine data views |
| /proj | | // EasyScreen configuring |

# 2.2 Editing files

## 2.2.1 Editing the file at the controller

Use can use a copy of the sample file in order to make specific adjustments. Template files are available in the following directories:

/**siemens**/sinumerik/hmi/template/cfg and /**siemens**/sinumerik/hmi/template/lng

Save the sample file in the "user" or "oem" directory - in the appropriate folder.

**Note**

As soon as the file is in the user-specific directory, the entries of this file have priority over the Siemens file. If an entry is missing in a user-specific file, the corresponding entry from the Siemens file is used instead.

You need to restart the operating software for the settings made in the file to take effect.

### Precondition

- If you copy files, you require access level 1 (manufacturer).
- When assigning a new file name, ensure that only files with a maximum name length of 49 characters can be managed.

### Copying / pasting / opening a file




1. Select the "Start-up" operating area.




2. Press the "System data" softkey.
   The data tree is displayed.
3. For instance, on the NCU directory, "System CF card", under "siemens" open the required directory (e.g. /sinumerik/hmi/template/cfg).

4. Position the cursor on the desired file.




5. Press the "Copy" softkey.

6. For instance, on the NCU in the "System CF card" directory, under "oem" or "user" open the required directory (e.g. /sinumerik/hmi/cfg), in which the copied file should be saved.




7. Press the "Paste" softkey.
   If a file of the same name already exists, you receive a message. You can overwrite or rename the file.




8. Press the "OK" softkey.




9. You can open the selected file in the editor by pressing the "Open" softkey.

   - OR -




   Press the <INPUT> key.

   - OR -

   Double-click the highlighted file.

## Rename file

1. Select the required file.



2. Press the ">>" and "Properties" softkeys.
   The "Properties of..." window opens.
   The following data is displayed:
   - Path:
   - Name:
   - Creation date and time:
   - Date of change and time:




3. To change the file name, place the cursor in the "Name" entry field and overwrite the name. Date of change and time are simultaneously updated.




4. Press the "OK" softkey to save the new name.

## Cut-out/delete file

1. Select the required file.




2. Press the "Cut" softkey.
   The file is copied into the buffer and at the same time deleted from the previous directory.
   As the file is located in the buffer, you can also insert the file into another directory.

- OR -



2. Press the ">>" and "Delete" softkeys.
   You receive an alarm and you can delete the file by pressing the "OK" softkey.
   Press the "Cancel" softkey to cancel the delete operation.

### Displaying the file in the preview window

1. Select the required file.



2. Press the ">>" and "Preview window" softkeys.
   The preview window is displayed in the lower area of the window together with the file contents.







Press the "Preview window" softkey again to close the window.

## 2.2.2 Editing a file externally

To create or edit an XML file on an external PC with Windows, use a text editor that supports the required "UTF-8" coding, e.g. "TextPad".

This means that you can enter or directly insert all characters of the unicode system without rewriting them using the keyboard.

If files with a different coding are saved, the special characters are not correctly displayed in the editor. In SINUMERIK Operate, there is no automatic conversion into the UTF-8 coding!

### Saving the XML file in UTF-8 coding

1. Select the "Save As" dialog box.
2. Set the character set to "UTF-8".

### Entering comments in an XML file

If you are entering comments to explain a program, you must keep the following in mind:

- A comment always begins with the character string: <!--
- A comment ends with the character string: -->

### Example

<!-- Work offset: -->

**Note**

In the comment itself, you can never use two minus signs one right after the other!

### Special characters

If you use characters in the text where the syntax of XML is of significance, then these must specifically formatted:

| Special characters | Notation |
|---|---|
| & | & |
| ' | ' |
| < | < |
| > | > |
| " | " |

### Comments in an ini file

If you enter a comment in an ini file, start the comment line with a semicolon.

### Copy file

Using the "WinSCP" or "Access MyMachine" program, copy the file from an external computer to the NCU via a network connection – or in the opposite direction from the NCU to an external PC.

For a detailed description, see the following sub-book:

### References

NCU Operating System (IM7): Backing up and restoring data and WinSCP and PuTTY service tools

### See also

Configuring alarm and message texts via alarm text files (Page 167)

# General settings

# 3

## 3.1 Changing the language

The operating software is available as standard in several languages.

### Selecting a language

You can define the language selection mode using the following machine data.

| MD9100 $MM_CHANGE_LANGUAGE_MODE | Language selection mode |
|---|---|
| = 1 | The user-interface language is specified via the "Language Selection" window (default value). |




1. Select the "Start-up" operating area.




2. Press the "Change language" softkey.
   The "Language Selection" window opens. The language most recently set is highlighted.
3. Switch the cursor to the required language.




4. Press the <INPUT> key.

   - OR -




   Press the "OK" softkey.
   The operating software switches to the language selected.

### Defining two languages

| MD9100 $MM_CHANGE_LANGUAGE_MODE | Language selection mode |
|---|---|
| = 2 | Selection of two languages. During operation, the "Change language" softkey can be used to toggle between these languages. |




1. Select the "Start-up" operating area.




2. Press the "HMI" softkey.



3. Press the ">>" softkey.




4. Press the "Language selection" softkey.
   The "Language Selection" window opens.
   All the installed languages are displayed in the fields "First language" and "Second language".
5. Select a language in each of these.
6. Press the <INPUT> key.




- OR -




Press the "OK" softkey.




During operation, you can switch between the two languages by pressing the "Change language" softkey.

## 3.2 Set date/time

You have the possibility of manually setting the date and time for the NCU (PLC and Linux) or PCU/PC (Windows). In so doing, you can select from various display formats.

### Formats of the date display

| Format | Example |
|---|---|
| d.M.yy | 3.1.09 |
| dd.MM.yy | 03.01.09 |
| dd.MM.yyyy | 03.01.2009 |
| d/M/yy | 3/01/09 |
| dd/MM/yy | 03/01/09 |
| dd/MM/yyyy | 03/01/2009 |
| M/d/yy | 1/3/09 |
| MM/dd/yy | 01/03/09 |
| MM/dd/yyyy | 01/03/2009 |
| yyyy/MM/dd | 2009/01/03 |

The following applies:

| Date | Description of the entry |
|---|---|
| d | Single or double digit day of the month (1 - 31). |
| dd | Double digit day of the month, with leading zero (01 - 31). |
| M | Single or double digit month of the year (1 - 12). |
| MM | Double digit month of the year, with leading zero (01 - 12). |
| yy | Double digit year (00 - 99). |
| yyyy | Four-digit year (e.g. 2009). |

## Time display formats

| Format | Example | Hour system |
|---|---|---|
| H:m:s AP | 1:2:4 PM | 24 hour system, with display of "AM" / "PM" |
| h:m:s ap | 1:2:8 pm | 12 hour system, with display of "am" / "pm" |
| HH:mm:ss AP | 13:02:08 PM | 24 hour system, with display of "AM" / "PM" |
| hh:mm:ss ap | 01:02:09 pm | 12 hour system, with display of "am" / "pm" |
| H:mm:ss AP | 1:02:09 PM | 24 hour system, with display of "AM" / "PM" |
| h:mm:ss ap | 1:02:09 pm | 12 hour system, with display of "am" / "pm" |
| h:mm:ss | 13:02:09 | 24 hour system |
| hh:mm:ss | 13:12:04 | 24 hour system |

The following applies:

| Time symbol: | Description of the entry | |
|---|---|---|
| h | Single or double digit hour (0 - 23 or 1 - 12 for am/pm). | |
| hh | Double digit hour, with leading zero (00 - 23 or 01 - 12 for am/pm). | |
| H | Single or double digit hour (0 - 23 for AM/PM). | |
| HH | Double digit hour, with leading zero (00 - 23 for AM/PM). | |
| m | Single or double digit minute (0 - 59). | |
| mm | Double digit minute, with leading zero (00 - 59). | |
| s | Single or double digit second (0 - 59). | |
| ss | Double digit second, with leading zero (00 - 59). | |
| AP | Display of the time of day: | AM or am = before midday or<br>PM or pm = after midday |

## Procedure




1. Select the "Start-up" operating area.




2. Press the "HMI" softkey.




3. Press the "Date/Time" softkey.
   The "Set Date and Time" window opens.




4. Select the required formats for the date and time with the <SELECT> key in the "Format" field.

5. For SINUMERIK Operate in the NCU: In the entry fields after "new" enter the actual date and time.
   For SINUMERIK Operate on PCU/PC: In the entry fields after "new NCU + PCU" enter the actual date and time.




6. Confirm the entry with the "OK" softkey.
   The new date and time details are accepted and output in the fields "actual NCU" and "actual PCU".

**Note**

You require administrator rights to set the date and time on the PCU/PC.

## 3.3 Using the time server

To ensure that the correct time information is always automatically available to the control, you can connect up to three time servers to the operating software. Make your settings in the "Set Date and Time" window.

**Note**

The "Use time server" setting is not available for SINUMERIK Operate on PCU/PC/IPC.

| NOTICE |
| --- |
| **Time synchronization errors**<br>Do not set the time synchronization on two different systems; this means:<br>Select the NTP server either in the PLC project via PROFINET IO (X150) or in the file "basesys.ini" on the NCU in section [LinuxBase], but not in both simultaneously, as an additional time zone can be specified on the NCU, and not in the PLC project. |

### References

Further information on the configuration file "basesys.ini" and the use of external time servers can be found in the following document:

NCU Operating System (IM7) Commissioning Manual, chapter "Configuring the system" > "Configuration file "basesys.ini"" > "Section [LinuxBase]"

### Requirement

You require as a minimum access level 3 (User) to use time servers.

### Checking the connection status

The icon on the right in field "Name / IP" shows the status of the server connection.

| Icon | Meaning |
| --- | --- |
|  | The connection status is unknown. The settings have not yet been processed or set. |
|  | You are connected to the time server. The system is successfully synchronizing times. |
|  | The system is still attempting to synchronize with a time server. |

**Note**

The system checks the time server connection cyclically. If a change is made, it may take a few seconds for the setting to take effect.

### Procedure




1. Select the "Setup" operating area.




2. Press the "HMI" softkey.




3. Press the "Date/Time" softkey.
   The "Set Date and Time" window opens.
4. Select the "Use time server" check box.
   The "Name / IP" input field is displayed.
5. In field "Name / IP," enter the DNS name or IP address of the time server.
   You can enter up to three time servers separated by blanks.




6. Press the "Set time server" softkey.
   If a connection to a time server exists, the settings will be synchronized with the system immediately. After that, a cyclic check is made as to whether the settings are effective. The "Set Date and Time" window remains open.




7. Confirm the entry with the "OK" softkey.
   The time server settings and the current times are taken over from the control.

**Note**

If the field "Use time server" is cleared again, the standard "UTC" time zone will apply again.

## 3.4 Use time zone

To ensure that the controller has the correct time information even in countries with different time zones, you can select the appropriate time zone.

The time zone set here is used by the entire system to translate Universal Time Coordinated (UTC) time into local time.

**Note**

UTC is the default.

### Procedure




1. Select the "Setup" operating area.




2. Press the "HMI" softkey.




3. Press the "Date/Time" softkey.
   The "Set Date and Time" window opens.
4. Set the check mark in the "Use time zone" field.
   The "Time zone" input field becomes active.
5. Enter your time zone in the "Time zone" field.




6. Confirm the entry with the "OK" softkey.
   The current time zone is applied by the controller.

**Note**

Changing the time zone can take a few seconds.

**Note**

**Zones without daylight saving time/standard time**

When changing the time zone, take into account the local daylight saving time/standard time. For example, regions like China or Japan have no daylight saving time/standard time, but Europe and the USA do.

### Examples

Hours are added or subtracted from the UTC time. You can set the following time zones, for example:

| | |
|---|---|
| Europe | Etc/Universal (UTC) |
| | Europe/Amsterdam (UTC-1:00) |
| | Europe/Berlin (UTC-1:00) |
| USA | US/Central (UTC+6:00) |
| | US/Michigan (UTC+5:00) |
| | US/Pacific (UTC+8:00) |
| China | Asia/Hong Kong (UTC-8:00) |
| Japan | Tokyo (UTC-9) |

## 3.5 Configuring Caps Lock

The "CAPSLOCK on" function ensures that text entered via an external keyboard is always entered in upper case and not in lower case.

The key behavior is set using the following display machine data.

### Setting

| MD9009 $MM_KEYBOARD_STATE | |
|---|---|
| = 0 | CAPSLOCK off (default value) |
| = 2 | CAPSLOCK on |

If the machine data is changed, the system must be rebooted.

## 3.6 Activating/deactivating the virtual keyboard

The virtual keyboard is configured in the file "slguiconfig.ini".

### Procedure

1. Copy the "slguiconfig.ini" file from the folder /**siemens**/sinumerik/hmi/template/cfg.
2. Paste the copy of the file into the following directory: /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg.
3. Open the file in the editor.
4. To activate or deactivate the keyboard, make the following settings:
   - Activating the virtual keyboard:
     In the section [TouchPanel], EnableTouch = true
     In the section [Keyboard], EnableVirtualKeyBoard = true
   - Deactivating the virtual keyboard:
     In the section [TouchPanel], EnableTouch = true
     In the section [Keyboard], EnableVirtualKeyBoard = false
5. Double click in the input field to display the keyboard.

## 3.7 Defining the access rights for the "HMI restart" softkey

### Display machine data

Using the following display machine data, set the access rights from which access level the "HMI restart" softkey is displayed.

For SINUMERIK Operate on PC or PCU, the "EXIT" softkey is displayed.

| MD9110_$MM_ACCESS_HMI_EXIT | Access level of the "EXIT" softkey |
|---|---|
| = 1 | Access level 1 (manufacturer), default value |

### Restart the HMI

Use the following procedure to restart SINUMERIK Operate:

Open the operating area, e.g. with the <MENU SELECT> key → "Menu step enable key" → "HMI restart"

## 3.8 Changing the skin design

### Displaying the machine data

You can change the skin design via the following item of machine data.

| MD9112 $MM_HMI_SKIN | | Skin design<br>The number of the skin is indicated |
|---|---|---|
| = 0 | Skin 0 (traditional) | |
| = 1 | Skin 1 (SINUMERIK Operate Generation 2) | |

After changing the machine data, the operating software must be restarted.

## 3.9 Setting the screensaver

### Defining the time until the screen is darkened (screensaver)

The duration (in minutes), after which the screen is automatically darkened if no keyboard key is pressed, can be specified using the display machine data MD9006 $MM_DISPLAY_SWITCH_OFF_INTERVAL.

| NOTICE |
|---|
| **Influence of screen darkening on the service life**<br>To extend its service life, the recommended duration for darkening a screen is 60 minutes. If you significantly reduce the duration, i.e. the screen is switched dark too often, this tends to shorten the screen service life. |

### Interface signal

The screen is darkened by setting an interface signal.

| DB 19 byte | Signals to the operator panel front (PLC → HMI) | |
|---|---|---|
| DBB0 | Bit 1 | Bit 0 |
| | Darken screen | Brighten screen |

The following applies when actively darkening the screen using the interface signal:

- It is no longer possible to light up the screen again from the keyboard.
- The first keystroke on the operator panel already triggers an operator action.

**Note**

In order to prevent accidental operator actions via the interface signal when the screen is darkened, we recommend disabling the keyboard at the same time.

## 3.10 Inserting a user-specific run up screen

### OEM-specific run up screen

You can replace the Siemens run up screen by your own run up screen.

Procedure:

1. Call your own run up screen "`splash.png`".

   **Note**

   The file name must not contain any lower case letters.

2. Save your own run up screen in one of the following directories:
   /**user**/sinumerik/hmi/ico/ico640 or
   /**oem**/sinumerik/hmi/ico/ico640 or
   /**addon**/sinumerik/hmi/ico/ico640

   **Note**

   The size of the screen is of no significance. A resolution of 640x480 pixels is always set.

## 3.11 Using user interfaces in parallel

It is possible to use SINUMERIK Operate in parallel on several user interfaces with one NCU. To avoid functional conflicts between the individual interfaces and the controlled system, adapt the following configuration file:

### "systemconfiguration.ini" file

Create the new "systemconfiguration.ini" configuration file in one of the two directories:

<Installation path>/**user**/sinumerik/hmi/cfg

<Installation path>/**oem**/sinumerik/hmi/cfg

In the configuration file, enter the section **[miscellaneous]** with the parameter `HMIFunctionMode` and the appropriate value. The following value specifications are possible:

| Value | Description |
|---|---|
| `PanelMode` or `<empty>` | The operating software and the system behave by default.<br>It can lead to errors in the system if SINUMERIK Operate is operated in parallel on another user interface. |
| `InterfaceMode` | SINUMERIK Operate is switched to passive in the system and does not log on as a regular user interface. The operating software:<br>• does not write to the PLC interface<br>• does not monitor PLC hardkeys<br>• does not monitor MMC commands<br>• does not monitor language selection via PLC<br>• does not monitor PLC commands<br>• does not monitor Ctrl-Energy<br>• does not monitor dark-ON via PLC<br>• does not set HMI-Ready in the PLC |
| `TerminalMode` | SINUMERIK Operate is used in the system as a tool loading station. The user interface behaves passively. |

### Example

For a CNC machine, two user interfaces are to be used in the same system with one NCU.

Setting of the main user interface:
```
[miscellaneous]
HMIFunctionMode=PanelMode
```

Setting of the secondary user interface:
```
[miscellaneous]
HMIFunctionMode=InterfaceMode
```

## 3.12 Setting how login data (credentials) are handled when archiving

If you generate a commissioning archive, then SINUMERIK Operate issues a dialog query as to whether all of the login data set up for the drives and OPC UA should or should not be saved to the archive. You parameterize the login data handling using the following display machine data:

### Display machine data

| MD9112 $MM_SAVE_CREDENTIALS | | Backing up access authorization for network drives when archiving |
|---|---|---|
| = 0 | Dialog is displayed (default setting) | |

| | |
|---|---|
| = 1 | No dialog, access authorizations are not saved |
| = 2 | No dialog, access authorizations are always saved |

**Note**

Using shortcut key CTRL+ALT+S, no login data is saved to the archive.

## 3.13 VNC remote connections

From SINUMERIK Operate, you have the possibility of operating external systems via VNC:

- Define a remote access, and link the selection via a softkey configured in the operating area.
- Define key forwarding to use key combinations in the VNC control area as an alternative.

The remote connection is started using a softkey configured in the operating area menu. If the side screen is activated, the VNC operating area can be additionally called using the navigation bar of the side screen.

**Note**

The VNC operating area only supports hextile encoding with color depths of 16 or 32 bits.

### Requirement

In the plant or company network, there is at least one system where a VNC server has been set up.

### Examples

The following usage scenarios are possible, for example:

- Connection to additional controls
- Connection to the CAD system
- Connection to tool measuring station
- Connection to other PCs

Make the remote connection settings using the "Setting up the operating area" window.

### General information

| Entry | | Meaning |
|---|---|---|
| **Operating area 1 - 24** | | |
| Type | VNC | Remote access to an external control or PC |
| | Empty | A VNC that has already been set up is deleted |

| Entry | | Meaning |
|---|---|---|
| Computer name | | Name or IP address of the external system |
| Password | | Password for logging on to the VNC server<br>The password is shown encrypted with "*". |
| Position information | Off<br>Top<br>Bottom | Display position of the VNC access information:<br>• Name of the system or IP address<br>• Port<br>• Connection status (e.g. for an incorrect connection) |
| Port | | Interface to the VNC server<br>The default port is 5900. |
| Operate header | On<br>Off | Display bar for operating area and message line active/inactive |

## Specifications for the configured softkey

| Entry | | Description |
|---|---|---|
| Access level | Key-operated switch, position 0<br>...<br>manufacturer | Assign access rights to the actual connection: from access level 7 (key operated switch position 0) to access level 1 (manufacturer). |
| Softkey text | | Two-line inscription text of the softkey<br>%n is accepted as a line separator.<br>If the first line is too long, then a line break is automatically inserted.<br>If a space is present, it is taken as a line separator.<br>For language-dependent softkey texts, the text ID is entered, which is used to search in the text file.<br>If nothing is specified in the entry field, then the symbolic name of the operating area, which is automatically assigned, is used as softkey text (e.g. AreaVNC_500). |
| Softkey icon | No icon | No icon is shown on the softkey. |
| | hd_vnc.png<br> | File names of the icon displayed on the softkey. |
| Text file | | File and context for the language-dependent softkey text. If nothing is specified in the input fields, the text appears on the softkey as was specified in the "Softkey text" input field. |
| Text context | | |

## Key transfer to the external system

Press the "Details" softkey to display the area "Key forwarding", and activate the appropriate checkbox.

By pressing the "Select menu" key, an additional window opens in which you can select the required shortcut key.

**Note**

Please note that depending on the operator panel being used, not all keys will be available.

| Designation | Key combination | Meaning |
|---|---|---|
| M | Shift+F10 | Calls the "Machine" operating area (M key). |
| Cancel alarm | Esc | Deletes alarms and messages. |
| Machine | Shift+F11 | Calls the "Machine" operating area (hard key). |
| Program manager | Home (numpad) | Calls the "Program Manager" operating area. |
| Channel | F11 | Switches further for multiple channels. |
| Program | End (Numpad) | Calls the "Program" operating area. |
| Alarm | Page Up (Numpad) | Calls the "Diagnosis" operating area. |
| Menu Select | F10 | Opens the operating area menu<br>You can change the shortcut key in the input field. When attempting to edit, the "Alternative shortcut key for Menu Select" dialog opens, allowing the following parameterization be made:<br>A - Z, F1 - F12, Home, End, Page Up, Page Down<br>AND combinable with SHIFT, CTRL, ALT |
| Help | F12 | Calls the context-sensitive online help for the selected window. |
| Offset | Page Down (Numpad) | Calls the "Parameter" operating area. |
| Custom | Shift+F12 | Calls the "Custom" operating area. |
| Next Window | Home | Toggles between the windows |

## Automatic key transfer

If the VNC operating area is selected, then the following keys and/or key combinations are directly transferred to the VNC server:

| Keys of the operator panel | Key combination |
|---|---|
| Horizontal softkeys (upper bar, HU) | F1 – F8 |
| Vertical softkeys (right bar, VR) | Shift+F1 - Shift+F8 |
| Horizontal softkeys (lower bar, HL) | Ctrl+F1 - Ctrl+F8 |
| Vertical softkeys (left bar, VL) | Shift+Ctrl+F1 - Shift+Ctrl+F8 |
| Recall | F9 |
| ETC switchover | Shift+F9 |

| Keys of the operator panel | Key combination |
|---|---|
| Extended horizontal softkeys (upper bar, HU) | F13 - F20 |
| Extended, vertical softkeys (right bar, VR) and right direct keys HT8 | Shift+F13 - Shift+F20 |
| Extended horizontal softkeys (lower bar, HL) | Ctrl+F13 - Ctrl+F20 |
| Extended, vertical softkeys (left bar, VL) and left direct keys HT8 | Shift+Ctrl+F13 - Shift+Ctrl+F20 |
| Lower and left-hand softkey bar (HL, VL) | (Shift)+Ctrl+F1 - (Shift)+Ctrl+F8 |

## Procedure




1. Select the "Start-up" operating area.







2. Press the "HMI" and "Oper. area menu" softkeys.
   The "Set up operating area" window opens.
3. Select the softkey that you want to configure.




4. Press softkey ">> Level" once or twice in order to configure operating areas 8 to 24.




5. To allow entry fields to be edited, press the "Change" softkey.
6. Make the necessary settings for the VNC remote connection.




7. Press the "Details" softkey if you want to parameterize additional key forwarding.
   You can define an alternative shortcut key for the Select menu.




8. Press the "OK" softkey to save the data.




   If you press the "Cancel" softkey, then all of the data that has not been activated is rejected.
9. Restart the control system to activate the configuration.

**Note**

**Reserved softkeys**

Softkeys reserved for the system are marked, and are not available for free configuration.

# 3.14 Access levels

## 3.14.1 Definition of access levels

Access to programs, data and functions is protected in a user-oriented hierarchical system of seven access levels. These are divided into

- Three password levels for manufacturer, service and users
- Four keyswitch positions for users

There are access levels 1 to 7 (see table below); where

- 1 is the highest and
- 7 is the lowest level.

Table 3-1 Access level concept

| Access level | Protected by | Area |
|---|---|---|
| 1 | Password: SUNRISE (default value) | Manufacturer |
| 2 | Password: EVENING (default value) | Service |
| 3 | Password: CUSTOMER (default value) | User |
| 4 | Keyswitch 3 | Programmer, machine setter |
| 5 | Keyswitch 2 | Qualified operator |
| 6 | Keyswitch 1 | Trained operator |
| 7 | Keyswitch 0 | Semi-skilled operator |

### Password security

**Note**

**Assigning secure passwords**

Observe the following rules when creating new passwords:

- When creating new passwords, never choose passwords that can be easily guessed, e.g. simple words, sequences of consecutive keys on the keyboard, etc.
- Passwords must always contain a combination of upper-case and lower-case letters as well as numbers and special characters. Passwords must comprise at least eight characters. PINS must comprise an arbitrary sequence of digits.
- Wherever possible and where it is supported by the IT systems, a password must always have a character sequence as complex as possible.

You can use a password management program to assist you when dealing with passwords. With its help, you can save passwords and PINs in encrypted form, manage them and generate secure passwords.

**Note**

**Change of password between SINUMERIK Operate and NC**

The access levels for SINUMERIK Operate and NC have been merged. Changing a password for SINUMERIK Operate simultaneously changes the relevant NC password and vice versa. It is important to note the following behavior:

- When a general NC reset is performed, no passwords are reset to the default passwords.
- Following a software upgrade, the SINUMERIK Operate passwords apply to the NC.
- Once changed, a password can no longer be reset.

## 3.14.2 Modifying the access levels password

### Overview

You can activate the access levels protected with passwords from the user interface.

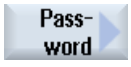
The following options are available for editing the passwords:

- Set new password
- Change password
- Delete password

### Setting the password




1. Select the "Start-up" operating area.




2. Press the "Password" softkey.




3. Press the "Set password" softkey.
   The "Set Password" window opens.




4. Enter one of the possible default passwords and press the "OK" softkey to confirm the entry.
   A permissible password is set and the valid access level is displayed. Invalid passwords will be rejected.

### Change password

You must change the standard passwords to obtain a secure access protection.




1. Press the "Change password" softkey.
   The "Change Password" window opens.
2. Select the area (e.g. user) for which you want to assign a new password.

3. Enter the new password in the input fields "New password" and "Repeat password".
   Always assign a secure password.




4. Press the "OK" softkey to confirm your entry.
   A new, valid password is only accepted if the two terms entered are identical.

**Note**

Invalid passwords are rejected by the operating software. Note the rules for strong passwords, for example:

- Minimum length of 8 characters
- At least one uppercase letter
- At least one lowercase letter
- At least one number
- At least one special character (ASCII 0x20 to 0x7E)

### Delete password




Press the "Delete password" softkey. The access authorization is reset.

**Note**

Access authorization is not automatically deleted at POWER ON.

## 3.14.3 Access levels for programs

### Objective

Access levels can be allocated to protect information; they can also be used to prevent unauthorized operation or deliberate damage to the control system or machine.

With access authorization levels to files and directories, it can be defined as to which users are authorized to execute which operations on a directory or a file in a file system. Operations in this sense include:

- Reading the contents of a file (READ), copying the file or placing the file in an archive.
- Changing the content of a file (WRITE) or deleting the file.
- Executing a file (EXECUTE).
- Listing and displaying a file in a directory (SHOW).
- Creating a file in a directory.

- Displaying or reading the contents of a directory.
- Creating or deleting a directory.

## Description of functions

The **individual allocation of access rights** acts, in the passive NC file system, on the following directories:

- Part programs
- Subprograms
- Manufacturer cycle directory
- User cycle directory
- Workpiece directory
- Workpiece directories

Further, the function acts on the files of the main program type (*.mpf) and subprograms or cycle (*.spf).

In addition, the technique can be expanded to include the following directories on the local drive:

- Part programs
- Subprograms
- Workpiece directory
- Workpiece directories

**Note**

**Working via remote access**

When working via remote access (e.g. WinSCP or Access MyMachine) the same access rights apply for editing or saving files and directories as directly at the controller.

## References

Refer to the following subbooks for access rights with WinSCP or Access MyMachine:

- NCU operating system, Chapter "Diagnostics and Service" > "System diagnostics" > "WinSCP and PuTTY"
- SINUMERIK Integrate Access MyMachine /OPC UA, Chapter "Introduction" > "Functions"

## Access rights via machine data

The following machine data is still available to **globally allocate access rights**:
MD51050 $MNS_ACCESS_WRITE_PROGRAM

Default setting = 4 (access level: User)

**Note**

**Effect of access rights**

The more restrictive setting is always effective both when individual access rights have been allocated and when settings have been made via MD51050 $MNS_ACCESS_WRITE_PROGRAM.

## Settings in the "slfsfileattributes.ini" file

The "slfsfileattributes.ini" file is available as a template in the following directory:

../siemens/sinumerik/hmi/template/cfg

The values from 1 ... 7 correspond to the access levels that, for example, also apply to the password. The 4-digit numbers designate the sequence according to the following rights:

| Significance from left to right: R W X S | | |
|---|---|---|
| R | READ | Reading files and directories |
| W | WRITE | Changing, deleting or creating files and directories |
| X | EXECUTE | Executes program: For files, for directories, always 7 |
| S | SHOW | Outputs a list of files and directories |

Copy the "slfsfileattributes.ini" file into one of the following directories:

../**oem**/sinumerik/hmi/cfg or ../**user**/sinumerik/hmi/cfg

The operating software must be restarted in order that the changes become effective.

## Example

The paths are composed of the data type identifiers. In the following example, "\wks.dir*.wpd *.mpf" comprises the data type for workpiece directories "wks.dir" the workpieces "*.wpd" and the part programs "*.mpf".

For all workpieces, 7777 is defined as default setting for new part programs.

All entries are in the section [ACCESSMASKS] and have the following form:

```
[ACCESSMASKS]

\wks.dir\*.wpd\*.mpf = 7777
\mpf.dir\*.mpf = 6577
\cus.dir\*.spf = 6577
```

## See also

Definition of access levels (Page 39)

## 3.14.4 Access rights for files

### Changing access rights

Access rights for execution, writing, listing and reading are displayed in the "Properties" window.

- Execute: is used for the selection for execution.
- Write: controls the changing and deletion of a file or a directory.

For NC files, you have the option to set the access rights from keyswitch 0 to the current access level, to be set separately for each file.

If an access level is higher than the current access level, it cannot be changed.

For external files (e.g. on a local drive), the access rights are displayed to you only if settings have been executed for these files by the machine manufacturer. They cannot be changed via the "Properties" window.

**Note**

**Settings for the access rights to directories and files**

Via the configuration file and MD51050, access rights of the directories and file types of the NC memory and user memory (local drive) can be changed and pre-assigned.

### Configuration file "slfsfileattributes.ini"

The access rights defined in the configuration file "slfsfileattributes.ini" and in MD51050 become valid in the NC when creating new files and can then be changed individually in the Program Manager using the "Properties" function. The access rights are an individual attribute of each file in the NC.

The access rights defined in the configuration file "slfsfileattributes.ini" and MD51050 become valid for external data on the CompactFlash Card of the NCU, or on the hard disk of the PCU/PC immediately for all the specified directories and file types, and cannot be changed through operation. The access rights can only be set globally for external data.

## 3.14.5 Access levels for softkeys

### Use

The display and operation of softkeys can be suppressed by both the manufacturer as well as the user. This allows the operating software to be specifically adapted to the required functional scope and therefore be configured as transparently as possible.

This means that the functional scope of the system is restricted, to prevent access to functions in the operating software, or to restrict the possibility of operator errors.

## Access hierarchy

A new access level can only be allocated for those softkeys that are also visible and can be used in the actual access level.

Example: If a softkey is only accessible with the "Manufacturer" access level, then the access level cannot be changed by an operator with access level "Key-operated switch setting 3".

## Restrictions

In order to guarantee that the system remains stable and operable, certain softkeys are excluded from the access level change, e.g. the "Customize softkeys" softkey itself, in order that the undo path is not cut-off or the softkey "<< Back" to the vertical softkey bar.

The following softkeys cannot be hidden or the access level changed:

- All "OK" softkeys
- All "Cancel" softkeys
- All "Accept" softkeys
- All "**<<**" Back and "**>>**" Continue softkeys

The access level for softkeys in the cycle dialog boxes can also be changed; this does not apply to softkeys in Easy Screen dialog boxes.

**Note**

**Software upgrade**

As all changes of the access levels for softkeys are described using the softkey ID, they still remain valid after a software upgrade, which may have resulted in softkeys being shifted.

Softkeys, where access is already protected by machine data - for example MD51045 $MNS_ACCESS_TEACH_IN - are only protected by the allocated access level after a new one has been allocated. The machine data is no longer evaluated.

All of the softkeys hidden using this mechanism can be temporarily displayed again using the access rights specified by the system. This is done using the "Show all softkeys" softkey. The softkey is deactivated as long as no softkeys have been allocated new access levels: This setting remains effective until the system is rebooted.

**Note**

**Applicability of modified access levels for softkeys**

The setting of specific access levels for softkeys on a PCU has an exclusive effect on the respective PCU softkeys themselves. To implement access rights on the NCU, both the manufacturer and the user must use the appropriate mechanisms and set the rights accordingly, see Chapter "Access levels for programs (Page 41)".

## 3.14.6 This is how you define new access levels for softkeys

### General sequence

In order to allocate a new access level to a softkey, proceed as follows:

- Activate the adaptation mode.
- Change the access level of a softkey.
- Accept the changes and confirm.

### Preconditions

Access to the "Customize softkeys" is protected using a general MD51073 $MNS_ACCESS_SET_SOFTKEY_ACCESS. This machine data is pre-assigned access level 3, i.e. the function is available from "Service" password and higher. In order that an operator with "User" access level can use this function, set this machine data to a value > 3.

An external mouse is required in order to make changes.

### Changing the access level of a softkey

#### Procedure

1. In the "Start-up" operating area, select the "HMI" menu.
2. On the expansion bar "**>>**" select the "Customize softkeys" softkey.
   You can obtain precise information about the procedure in the following dialog.
3. Confirm with "OK", in order to activate the "Customize the softkey access levels active" mode. A table is displayed.
4. Select an operating area.
5. Place the mouse cursor on a softkey and using the right-hand mouse key call the context menu "Define access level", in order to define a new access level for this softkey.
   - The displayed access level is the default setting.
   - A softkey is practically made invisible with "Always hide".
   - You can reset the additionally allocated access level using "Original setting". This selection is only listed after an individual access level was set.
6. In order to accept changes, return to the "Start-up" operating area and there, press the "Close customizing" softkey. This means that you exit the "Customize the softkey access levels active" mode.
   If you acknowledge the following query with "OK", then the changes are accepted and become effective once the system has rebooted.
7. If the changes to a softkey are not to be applied, select the line in the "Adapt softkey access levels" window and press the "Original setting" softkey. The identifier "Original setting" is entered in the "New access level" column. After the end of the softkey adaptation and system restart, the original setting of the softkey is restored and the line entry in the "Adapt softkey access levels" window is deleted.

### Example

The following access levels were changed:

| Softkey | New access level |
|---|---|
| SISuMainScreenMenuHu::0::3 | Manufacturer |
| SIMaJogMillMenuHU::jogHuMore::6 | User |
| SISuPaUserDataMenu::::4 | Service |

Once selected, the identifier and position of the softkey are transferred to the "Softkey" column.

### Result

When the changes have been accepted, a configuration file with the name "slsoftkeyaccess.xml" is created and saved in the file system under the following path: ../**user**/sinumerik/hmi/cfg. Files can also be saved on the **/oem** path; these files are evaluated when the system boots. The following priority applies: oem < user.

**Note**

**Editing the file "slsoftkeyaccess.xml"**

In order to avoid errors in the xml syntax and for the softkey identifier the "slsoftkeyaccess.xml" file may only be edited via the operating software as described above, and not directly in the file system under "System data". It is possible that the complete file or individual entries for softkeys can no longer be interpreted.

## 3.14.7 Setting the MDA program access via protection level

Use the following general machine data to set the protection level from which programs can be edited in MDA operating mode. The setting also applies to program overwriting in Automatic mode. The access levels 0 to 7 are possible.

### Setting the protection level

| MD51075 $MNS_ACCESS_WRITE_MDI | Write protection level MDA program |
|---|---|
| = 7 | No interlock, editing is permitted (standard) |

**Note**

A change of the protection level is activated after POWER ON.

If the current access level is not sufficient, a program under MDA mode can neither be edited nor changed using the "Load MDA" and "Delete blocks" softkeys.

# 3.15 Program selection

## 3.15.1 Displaying the softkey

The softkeys on the user interface for calling the program lists are activated via general configuration machine data.

### Configuring a softkey

| MD51041 $MNS_ENABLE_PROGLIST_USER | |
|---|---|
| 0 | No softkey |
| 1 | The "Prog. list" softkey is displayed |

| MD51043 $MNS_ENABLE_PROGLIST_MANUFACT | |
|---|---|
| 0 | No softkey |
| 1 | The "MANUFACTURER" softkey is displayed |

### Precondition

To create program lists, you require the following authorization:

- "Program manager" operating area: Access level 3 (password: User) or
- "Start-up" operating area: Access level 1 (password: Manufacturer).

### Program lists

You can create a program list by combining desired programs from the NC data, e.g. workpieces, part programs and subprograms. These programs can then be selected by the PLC for execution.

| Program lists | Directory |
|---|---|
| plc_proglist_user.ppl | /**user**/sinumerik/hmi/plc/programlist |
| plc_proglist_manufacturer.ppl | /**oem**/sinumerik/hmi/plc/programlist |

### Procedure




1. Select the "Start-up" operating area if you have the authorization for access level 1 (password: Manufacturer).

   - OR -




Select the "Program manager" operating area if you have the authorization for access level 3 (password: User).



2. Press the menu forward key and the "Prog. list" softkey.
   The "Program List" window opens.







3. Press the "Manufacturer" softkey if you want to create a program list that can be used for all machines of a certain series.
4. Place the cursor in the desired line (program number).




5. Press the "Select program" softkey.
   The "Programs" window opens. The data tree of the NC memory with the directories "Workpieces", "Part programs" and "Subprograms" is displayed.




6. Place the cursor on the desired program and press the "OK" softkey.
   The selected program is inserted in the first line of the list together with its path.
   - OR -
   Enter the program name directly in the list.
   If you are making entries manually, check that the path is correct (e.g. //NC/WKS.DIR/MEINPROGRAMM.WPD/MEINPROGRAMM.MPF).
   If required, the extension (.MPF) is added.
   **Note:**
   A syntax check is not performed.
   With workpieces, make sure that there is a main program or a job list with the same name in the workpiece itself.




7. To remove a program from the list, place the cursor on the appropriate line and press the "Delete" softkey.
   - OR -




To delete all programs from the program list, press the "Delete all" softkey.

# 3.16 Activate and configure the Sidescreen window

## 3.16.1 Activate Sidescreen

In addition to SINUMERIK Operate, Multitouch Panels with widescreen displays offer the option of using the additional area at the side of the screen for displaying sidescreen windows. The Sidescreen windows give you a quick and permanent view of important information and enable you to operate virtual keys for quick selection. In addition, you can also configure your own windows.

### Precondition

To use the function, you need a Multitouch panel in widescreen format (1366x768, 1920x1080 resolution).

### Activation

You activate the Sidescreen function via the following display machine data:

| MD9114 $MM_SIDESCREEN | SINUMERIK Operate Sidescreen |
|---|---|
| = 1 | Enables the additional Sidescreen display area of the Multitouch Panel (HD Ready or Full HD). |
| = 2 | Enables the Sidescreen display area in HD Ready.<br>Enables the Display Manager (Workbench) in Full HD. |

**Note**

The setting with the value = 2 is only possible with a display size of 22 inches.

### Axis load display

You activate the detection of additional drive actual values for the axis load display in the Sidescreen:

| MD36730 $MA_DRIVE_SIGNAL_TRACKING | Detection of additional drive actual values |
|---|---|
| = 1 | $AA_LOAD: Drive load<br>$AA_POWER: Drive efficiency<br>$AA_TORQUE: Drive torque setpoint<br>$AA_CURR: Smoothed current actual value (quadrature-axis current) of the drive |

**Note**

Detection for the axis load display is only possible if actual values are available from the drive.

### Bar display for axis and spindle power

You can assign parameters for the bar display for the axis power using the following axis-machine data:

| MD53030 $MAS_AXIS_MAX_POWER | Maximum value of the axis power display |
|---|---|
| = 100 | Default<br>Maximum permitted axis power in percent<br>The bar display is green in the range between 0 and the value of AXIS_MAX_POWER. |

| MD53031 $MAS_AXIS_POWER_RANGE | Display range of the axis power display |
|---|---|
| = 100 | Default<br>Full-scale value for axis power display in percent<br>This value must be greater than or equal to AXIS_MAX_POWER. The bar display is red in the range between the values of AXIS_MAX_POWER and AXIS_POWER_RANGE. |

You can assign parameters for the bar display for the spindle power using the following general machine data:

| MD51030 $MNS_SPIND_MAX_POWER | Maximum value of the spindle power display |
|---|---|
| = 100 | Default<br>Maximum permitted spindle power in percent<br>The bar display is green in the range between 0 and the value of SPIND_MAX_POWER. |

| MD51031 $MNS_SPIND_POWER_RANGE | Display range of the spindle power display |
|---|---|
| = 100 | Default<br>Full-scale value for spindle power display in percent<br>The value must be equal to or greater than SPIND_MAX_POWER. The bar display is red in the range between the values of SPIND_MAX_POWER and SPIND_POWER_RANGE. |

## 3.16.2 Display Sidescreen ABC keyboard

Not only the standard windows but also the window for an ABC keyboard can be displayed in the Sidescreen display of the Multitouch panel.

### Procedure

1. Copy the sample file "slsidescreen.ini" from the directory /siemens/sinumerik/hmi/template/sidescreen
2. Place the copy of the file in one of the following directories:
   /**oem**/sinumerik/hmi/cfg or
   /**user**/sinumerik/hmi/cfg

3. Open the copy and delete the comment sign ";" in front of the keyword `PAGE002`.

   **Note**

   The values of the attributes `name` and `implementation` must not be changed.

4. Save the setting.
   The ABC keyboard can be selected after a run-up.

## 3.16.3 Display Sidescreen machine control panel

Not only the standard windows but also the window for machine control panel functions can be displayed in the Sidescreen display of the Multitouch panel.

The example of a machine control panel is displayed by the following procedure:

### Procedure

1. Copy the sample files "slsidescreenmcppageconfig.xml", "slsidescreenmcppage_eng.ts" and the subdirectories of the icon files from the directory
   /siemens/sinumerik/hmi/template/sidescreen/mcp/
   The icon files are stored in the following subdirectories:
   /ico1024 for HD Ready
   /ico1920 for Full HD
2. Place the copy of "slsidescreenmcppageconfig.xml" in one of the following directories:
   /**oem**/sinumerik/hmi/proj or
   /**user**/sinumerik/hmi/proj
3. Place the copy of "slsidescreenmcppage_eng.ts" in one of the following directories:
   /**oem**/sinumerik/hmi/lng or
   /**user**/sinumerik/hmi/lng

   **Note**

   The TS file is automatically integrated by the system during startup.

4. Save the icon files under the following path in one of the intended subdirectories:
   /**oem**/sinumerik/hmi/ico or
   /**user**/sinumerik/hmi/ico
   Depending on the screen resolution, there are the following subdirectories:
   /ico1024 for HD Ready
   /ico1920 for Full HD
5. Copy the sample file "slsidescreen.ini" from the directory
   /siemens/sinumerik/hmi/template/sidescreen
6. Place the copy of "slsidescreen.ini" in one of the following directories:
   /**oem**/sinumerik/hmi/cfg or
   /**user**/sinumerik/hmi/cfg

7. Open the copy of the file "slsidescreen.ini", and delete the comment sign ";" in front of the keyword `PAGE003`.

   **Note**

   The values of the attributes `name` and `implementation` must not be changed.

8. Save the setting.
   The machine control panel can be selected after a run-up.

### 3.16.3.1 Configuring your own keys on the machine control panel

Own keys can also be displayed in the Sidescreen for the machine control panel. To do this, the keys must be configured in the file "slsidescreenmcppageconfig.xml". The properties, position of the keys and the headers are specified here.

#### Example of the file "slsidescreenmcppageconfig.xml"

```
<sideScreenControlPanel>
 <mainBlock id="sideScreenMcpBlock">
  <block position="1" title="SPINDLE">
   <row position="1">
    <buttonGroup spacing="false" title="COOLANT">
     <button position="1" text="ON" icon="slsidescreenmcp_cooling.png"
style="redStyle" title="Spindle">
      <onPressed address="HmiUserKey1"/>
      <active address="HmiUserLed1"/>
      <disabled address="DisableHmiUserKey1"/>
     </button>
    <button position="2" text="OFF" icon="slsidescreenmcp_cooling_off.png">
      <onPressed address="HmiUserKey2"/>
      <active address="HmiUserLed2"/>
     </button>
    </buttonGroup>
   </row>
  </block>
 </mainBlock>
</sideScreenControlPanel>
```

#### Adaptation options of the file "slsidescreenmcppageconfig.xml"

You can make the settings for your own keys in the `<sideScreenControlPanel>` tag:

| Tag | Description | |
|---|---|---|
| `<mainBlock id="sideScreenMcpBlock">` | Is the main section in the Sidescreen display | |
| `<block position="1" title="SPINDLE">` | A main section has any number of blocks. | |
| | `position` | Order of the blocks |
| | `title` | Block header |
| `<row position="1">` | Each block has any number of lines. | |
| | `position` | Positioning of the lines |

| Tag | Description | |
|---|---|---|
| <buttonGroup spacing="false" title="COOLANT"> | A key group combines two keys under one header. | |
| | spacing | The property enables the display type of the keys to be determined: together "false" or separate "true". The default is separate. |
| | title | All the keys in a key group have the same header. |
| <button position="1" text="ON" icon="slsidescreenmcp_cooling.png" style="redStyle" title="Spindle"> | Defines the key properties | |
| | position | Position of the key |
| | text | Labeling shown on the key |
| | icon | Icon of the key<br>The icon is displayed above the text label. If there is no labeling, then the icon also takes up the labeling space. |
| | style | Changes the background color assigned to the key<br>Possible values are:<br>"redStyle" for a red key<br>"greenStyle" for a green key<br>The default is "defaultStyle" |
| | title | A key that does not belong to any group can have its own header. |
| <onPressed address="HmiUserKeyX"/> | If the key is pressed, then the PLC bit is set to 1.<br>The value remains at 1 as long as the key remains actuated. With a brief press, the value remains at 1 for at least 300 ms. | |
| | address | Alias name "HmiUserKeyX" for the addressed PLC bit<br>Values of 1 to 64 are permitted for X.<br>HmiUserKey1 to HmiUserKey64 stand for DB19.DBB404.0 to DB19.DBB411.7. |
| <active address="HmiUserLedX"/> | The key is shown as active (LED animation at the top edge of the key). | |
| | address | Alias name of the PLC bit from which the active identifier is derived<br>Values of 1 to 64 are permitted for X.<br>HmiUserLed1 to HmiUserLed64 stand for DB19.DBB412.0 to DB19.DBB419.7. |
| <disabled address= "DisableHmiUserKeyX"/> | The key is shown as inactive (icon and text grayed out). | |
| | address | Alias name of the PLC bit from which the inactive identifier is derived.<br>Values of 1 to 64 are permitted for X.<br>DisableHmiUserKey1 to DisableHmiUserKey64 stand for DB19.DB420.0 to DB19.DB427.7. |

### Store own icons

You can create your own icon files under one of the following paths:

/**oem**/sinumerik/hmi/ico or

/**user**/sinumerik/hmi/ico

Depending on the screen resolution, under the following subdirectories:

/ico1024 for HD Ready

/ico1920 for Full HD

**Note**

The icon file names must be the same as the names used in the XML file, e.g. `icon="slsidescreenmcp_cooling.png"`.

## 3.16.3.2 Create own translations of texts for the machine control panel

You can display all machine control panel texts in different languages. Translations are created in the file "slsidescreenmcppage_xxx.ts".

"xxx" stands for a language code, e.g. "deu" for German, "eng" for English, and so on. You will find the supported languages at "Supported languages (Page 502)"

### Procedure

1. Create the file "slsidescreenmcppage_xxx.ts" with the desired language code in one of the following directories:
   /**oem**/sinumerik/hmi/lng or
   /**user**/sinumerik/hmi/lng
2. Open the language file and insert the designations accordingly.
   The following texts may be displayed in translation:
   - All headers, e.g. `title="COOLANT"`
   - All key labels e.g. `text="ON"`
3. Save the language file.
   The translations are displayed after a run-up and the language has been changed.

### Adaptation options of the file "slsidescreenmcppage_xxx.ts"

You can insert the translations in the `<message>` tag:

| Tag | Description |
|---|---|
| `<source>ON</source>` | This is the language code of the original language. |
| `<translation>ON</translation>` | This is the translation. |
| `<chars>8</chars>` | This is the maximum length of the translation.<br>The tag serves merely a note for the translator. |

### Example extract of the file "slsidescreenmcppage_eng.ts"

```
...
<message>
 <source>ON</source>
 <translation>EIN</translation>
 <chars>8</chars>
</message>
...
```

# 3.17 SINUMERIK Operate Display Manager

## 3.17.1 Display configurations

Using the Display Manager and display configurations, it is possible to reduce SINUMERIK Operate, in other words, run in less than full-screen mode, and to allocate the vacant space to other applications. These applications are either standard Windows applications (.EXE) or applications that belong to the Display Manager. Both types of applications can also be developed by third parties using the OA Toolkit from SINUMERIK Operate and integrated into a display configuration. To display the applications, the screen is divided into "frames". Each frame may contain an application or a menu for controlling an application displayed in another frame. Several screen segments or "Displays" are possible and you can navigate between them during the program runtime. These displays are combined into a display configuration. There may be multiple display configurations, for example, for different screen resolutions.

A display configuration is defined for a specific screen resolution. The available display configurations must be made known to the system in the `[displayConfigurations]` section of the `systemconfguration.ini` file. The display configuration itself is described in a separate configuration file.

### Example of the "systemconfiguration.ini" file

```
[displayConfigurations]
DC001 = res:=1366x768, file:=displayconfig_1366x768
DC002 = res:=1920x1080, file:=displayconfig_1920x1080
```

In this example, there are two display configurations for the resolutions 1920x1080 and 1366x768. The names of the entries always begin with the string "DC" followed by a sequential number. Each entry has a "file" property that points to the file with the display configuration. The name of the file is arbitrary. Similar to other configuration files, it is also searched for in the cfg directories under oem and user. The ".ini" file extension does not have to be specified. The "res" property indicates the resolution to which the respective display configuration applies.

If there is no display configuration for a resolution, the user interface is displayed full screen at this resolution.

### Activation

To activate the SINUMERIK Operate Display Manager, the following display machine data must be set.

| MD9114 $MM_SIDESCREEN | SINUMERIK Operate Sidescreen |
|---|---|
| = 2 | Enables the Sidescreen display area in HD Ready.<br>Releases the Display Manager in Full HD. |

The default configurations used by the Display Manager for the two screen resolutions, 1920x1080 and 1366x768, are in the two files:

- sldm_1920x1080.ini
- sldm_1366x768.ini

These files can be used as a template for custom display configurations. This section describes how to create custom display configurations.

## 3.17.2 Creating your own display configurations

### 3.17.2.1 Displays

One or more displays can be defined in the respective configuration file for each display configuration. A name and a list of frames are specified for each of these displays. The frames describe the division of the display into rectangular areas in which applications or menus can be displayed.

### Example

```
[displays]
DISPLAY001= name:=display_1, frames:="frame1, frame2, frame3"
DISPLAY002= name:=display_2, frames:="frame4, frame5, frame6, frame7"
…
```

| Property | Meaning |
|---|---|
| `name` | Name of the display. Required, for example, for switching displays |
| `frames` | Contains a comma-separated list of frames that should be present on this display to describe the layout of the display |
| `enableMirrorMode (true\|false)` | Activates (true) or deactivates (false) mirror mode for the display<br>When the mirror mode is activated, all frames of the display are displayed mirrored horizontally when the setMirrorMode (true) function is called. Otherwise, the appearance of the frames remains unchanged after calling the setMirrorMode (true) function.<br>"true" is default. |

| Property | Meaning |
|---|---|
| onStartup | Contains a semicolon-separated list of actions to be performed when the display first appears<br>The possible actions are described below. |
| onShow | Contains a semicolon-separated list of actions to be performed each time the display fades<br>The possible actions are described below. |

### 3.17.2.2 Frames

A frame is a container for applications or menus and primarily describes the desired position and size of the application and menus in pixels. In addition, this is also where the application or menu that should be displayed in the frame after the Display Manager starts is specified. The applications that can be assigned to the frames are specified in the systemconfiguration.ini file.

#### Example

```
[frames]
FRAME001= name:=frame1, x:=0, y:=0, width:=460, height:=970,
app:=mydialog
FRAME002= name:=frame2, x:=460, y:=0, width:=1460, height:=1080,
app:=OPERATE
FRAME003= name:=frame3, x:=0, y:=970, width:=460, height:=110,
menu:=mymenu
FRAME004= name:=frame4, x:=0, y:=0, width:=1280, height:=970,
app:=mydialog2, runableApps:="mydialog, mydialog2"
FRAME005= name:=frame5, x:=0, y:=970, width:=1280, height:=110,
menu:=menu2
FRAME006= name:=frame6, x:=1280, y:=0, width:=640, height:=480,
app:=OPERATE
FRAME007= name:=frame7, x:=1280, y:=480, width:=640, height:=600,
app:=mydialog3
…
```

If SINUMRIK Operate is to be displayed in a frame, the application name is specified as "OPERATE". It should be noted that SINUMERIK Operate can only be used in certain sizes. The currently supported sizes (in pixels) are:

- 640x480
- 760x505
- 800x480
- 800x600
- 1024x768
- 1024x1014
- 1080x810
- 1280x768
- 1280x800

- 1280x1024
- 1300x768
- 1366x768
- 1440x1080
- 1454x1080
- 1854x1014
- 1854x1080
- 1920x1080

| Property | Meaning |
|---|---|
| `name` | Name of the frame. Used in the frame list of the displays, for example |
| `x, y, width, height` | Absolute position (x, y), width and height of the frame in pixels |
| `app` | Application displayed in the frame the first time the frame is displayed. |
| `runableApps` | List of applications that may be displayed in this frame in case an application is swapped. |
| `menu` | Menu that is displayed in the frame.<br>Ignored if "app" is used at the same time |

### 3.17.2.3 Menus

In addition to applications, menus can also be displayed in the frames of a display. These menus are typically used to display certain applications in another frame or to switch displays.

A menu configuration consists of four parts:

- Description of the individual menu items (section `[menuitems]`)
- Description of the look and feel of the menu items (section `[menuitemstyle]`)
- Description of the menu structure (section `[menus]`)
- Description of the look and feel of the menu (section `[menustyle]`)

The `[menuitems]` section defines the menu items (keys/buttons). Each entry corresponds to a key/button. The properties of the menu items can be configured.

The following properties are available:

| Property | Meaning |
|---|---|
| `name` | Unique name of the menu item. |
| `onClicked` | Action to be triggered by clicking on this menu item. The actions that are possible are described in the "Menu actions" section. |
| `text` | Text displayed on the menu item. Line breaks can be generated with `\n` or `%n`. |
| `textID` | A text ID that references a language-dependent text. If the text can be read, it replaces the text specified in the "text" property. |
| `textContext` | The context associated with the text ID. The context can also be specified when configuring a menu or globally on the file level. Specifying a context in the menu item overwrites any existing context information in the menu or on the file level. Specifying a context in the menu overwrites any existing context information on the file level. |
| `image` | File name of the icon that is displayed on the menu item.<br>Special case for display button:<br>For buttons that are used to switch displays, the icon to be displayed on the button can be generated automatically by the Display Manager. To do this, you specify the name of the display to be switched with the file extension ".auto" as the file name. A schematic drawing of the display is then displayed as an icon. The stroke width and stroke color used for this representation can be specified with the properties strokeWidth and strokeColor.<br>Display buttons have no text and no "imagePressed" icon. |
| `strokeWidth`[1)] | Stoke width of the schematic drawing for display buttons (see Property image). Default: 1. |
| `strokeColor`[1)] | Color of the stokes of the schematic drawing for display buttons (see Property image).<br>Default: Qt::lightGray |
| `imagePressed` | File name of the icon displayed on the menu item in the pressed state. As an alternative to specifying this property, a file named "<image>_activated" is automatically used, if available; <image> is the name specified in the "image" property. |

| Property | Meaning |
|---|---|
| accessLevel | Access level from which the menu item is displayed:<br>System, Manufacturer, Service, Customer, Key_3, Key_2, Key_1, Key_0 or numerical equivalent (0-7)<br>Default: Key_0 (7) |
| menuItemStyle | Style of the menu item that defines all visual properties of the menu item. |
| | **Note**<br>All of the properties described below can be specified for configuring the respective menu items as well as for defining the MenuItemStyles. An entry in the menu item overwrites the corresponding entry in the MenuItemStyle. |
| textSize[2)] | Font size |
| textAlignment[2)] | Alignment of the text of the menu item:<br>Left, Right, Top, Bottom, TopLeft, TopRight, BottomLeft, BottomRight (analog Qt::Alignment)<br>You can also specify the values of the Qt flag field in hex (0x00) or decimal.<br>Default: Qt::AlignCenter |
| imageAlignment[2)] | Alignment of the icon of the menu item:<br>Left, Right, Top, Bottom, TopLeft, TopRight, BottomLeft, BottomRight (analog Qt::Alignment)<br>You can also specify the values of the Qt flag field in hex (0x00) or decimal.<br>Default: Qt::AlignCenter |
| width[2)] | Width of the menu item in pixels.<br>Default behavior:<br>The menu items equally share the available space in the menu (horizontal menu) less spacing, or are as wide as the menu minus the margin (column menu).<br>For grid menus, the menu items equally share height and width. |
| height[2)] | Height of the menu item in pixels.<br>Default behavior:<br>The menu items equally share the available space in the menu (vertical menu) less spacing, or are as tall as the menu minus the margin (row menu).<br>For grid menus, the menu items equally share height and width. |
| color[2)] | Color of the menu item.<br>See paragraph "Color definitions".<br>Default: Qt::lightGray |
| pressedColor[2)] | Color of the menu item in the pressed state.<br>See paragraph "Color definitions".<br>Default: Qt::blue |
| textColor[2)] | Color of the menu item text<br>See paragraph "Color definitions".<br>Default: Qt::black |

| Property | Meaning |
|---|---|
| pressedTextColor[2] | Color of the menu item text in the pressed state of the menu item<br>See paragraph "Color definitions".<br>Default: Qt::white |
| borderWidth[2] | Width of the border of the menu item in pixels<br>Default: 1. |
| borderColor[2] | Color of the border of the menu item<br>See paragraph "Color definitions".<br>Default: Qt::darkGray |
| textRect[2] | Rectangle relative to the menu item (widget coordinate system, i.e. Top-Left of the menu item is coordinate 0,0) in which the text is aligned.<br>The following should be specified:<br>x, y, width, height. For example, textRect:="0,10,40,30" or textRect:=0/10/40/30 or textRect:=0 10 40 30<br>Note the required quotes when commas are used as delimiters, because commas are also the separator between the properties.<br>Default: The rectangle within the border; for borderWidth = 0 this is the rectangle of the complete menu item. |
| imageRect[2] | Rectangle relative to the menu item (widget coordinate system, i.e. Top-Left of the menu item is coordinate 0,0) in which the icon is aligned.<br>The following should be specified:<br>x, y, width, height. For example, imageRect:="0,10,40,30" or imageRect:=0/10/40/30 or imageRect:=0 10 40 30<br>Note the required quotes when commas are used as delimiters, because commas are also the separator between the properties.<br>Default: The rectangle within the border; for borderWidth = 0 this is the rectangle of the complete menu item. |

[1] Only relevant for display buttons

[2] Applies to menuItem and menuItemStyle

## Example 1

```
[menuitems]
MENUITEM001= name:=menuitem1, onClicked:=showDisplay(display1),
text:="Display1"
MENUITEM002= name:=menuitem2, onClicked:=showDisplay(display2),
text:="Display2", accessLevel:=Customer, width:=100
MENUITEM003= name:=miPdf, menuItemStyle:=misMenu,
onClicked:="showApp(defaultFrame, appPdf)", image:=dm_pdf.png,
text:="PDF"
MENUITEM004= name:=miBrowser, menuItemStyle:=misMenu,
onClicked:="showApp(defaultFrame, appBrowser)",
image:=dm_browser.png, textID:=TEXT_ID_BROWSER
```

In the `[menuitemstyles]` section, styles can be defined for menu items in which all visual properties of a menu item are defined. The individual properties can in turn be overwritten in the menu item. The summary of all visual attributes saves typing work, because you usually have many menu items that generally should all look the same. If a property should then be different for a menu item, it can be overwritten individually.

Description of MenuItemStyle properties: See above.

## Example 2

```
[menuitemstyles]
MENUITEMSTYLE001= name:=misMenu, textSize:=10,
textAlignment:=bottom, imageAlignment:=top, color:=37/53/63,
textColor:=153/173/185, pressedColor:=52/87/140,
pressedTextColor:=White, borderColor:=37/53/63,
pressedBorderColor:=52/87/140, borderWidth:=5, height:=66, width:=66
```

For unspecified properties, the corresponding default values (see above) take effect.

In the `[menus]` section, one or more `menuItems` are combined into a menu. This is done using the `menuItems` property, which contains a list of the menu items (buttons/buttons) to be displayed in a certain menu. The available properties are listed in the following table:

| Property | Meaning |
|---|---|
| `menuItems` | List of menu items.<br>Comma-separated list of menu item names<br>With the "grid" menu layout, the menu items are specified as follows to determine their position within the menu:<br>menuItem(row, col, rowSpan=1, colSpan=1)<br>The indexes "row" and "col" start with the value 0.<br>Specifying "rowSpan" and "colSpan" is optional, default value is 1 (analogous to QGridLayout::addWidget ())<br>With the "row" or "col" menu layout, you can also use the specification of spacing (pixel) and stretch (factor = 0):<br>• Spacing inserts "int" pixels of additional spacing between the menu items (analogous to QBoxLayout::addSpacing ())<br>• Stretch inserts a stretchable additional distance between the menu items, an expansion factor can be optionally specified (analogous to QBoxLayout::addStretch ()) |
| `defaultFrame` | Name of the frame on which the menu items of the menu are to act and that use the "defaultFrame" string as target specification in the showApp statement. This allows menu items to be reused in multiple menus. |
| `textContext` | The context for the text ID of the menu items (see above). |
| `itemAccessLevel` / `accessLevel` | Access level from which the menu items are displayed:<br>System, Manufacturer, Service, Customer, Key_3, Key_2, Key_1, Key_0 or numerical equivalent (0-7)<br>Default: Key_0 (7) |
| `menuStyle` | Style of the menu.<br>Sets all visual properties |
| | **Note**<br>All of the properties described below can be specified for configuring the respective menus as well as for defining the MenuStyles. An entry in the menu overwrites the corresponding entry in the MenuStyle. |

| Property | Meaning |
|---|---|
| layout[1)] | Alignment of menu items:<br>"row", "col" or "grid". With a "row" or "col" layout, the menu items are aligned horizontally in a row or vertically in a column.<br>With a "grid" layout, the menu items are arranged in a grid.<br>Default: "row" |
| margin[1)] | Margin between the menu items and the outer edge of the menu.<br>Value in pixels, (analogous to QBoxLayout::setMargin() or QGridLayout::setContentsMargin() (all 4 directions))<br>Default: 1. |
| spacing[1)] | Spacing between the menu items.<br>Value in pixels, (analogous to QBoxLayout::setSpacing() or QGridLayout::setSpacing())<br>Default: 1. |
| color/ backgroundColor[1)] | Color (background color) of the menu.<br>See paragraph "Color definitions".<br>Default: Qt::gray |

[1)] Applies to menu and menuStyle

## Example 3

```
[menus]
MENU001= name:=menu1, menuItems:="menuitem1, menuitem2, menuitem3"
MENU002= name:=menu2, orientation:=vertical, margin:=2, spacing:=2,
items:="menuitem4, menuitem5, spacing(50), menuitem6, menuitem7,
stretch()", itemHeight:=100
MENU003= name:=mMenuOben, menuStyle:=msMenu, defaultFrame:=fTop,
items:="miPdf, miBrowser, spacing(10), miOperate, stretch(),
miMaximize"
MENU004= name:=mGridMenu, menuStyle:=msGrid, defaultFrame:=fBottom,
items:="miMachine(0,0), miProgram(0,1), miProgramManager(0,2),
miDiagnosis(0,3), miPdf(1,0), miBrowser(1,1), mOperate(1,2,1,2),
miMirror(0,3,2,1)"
```

In the [menustyles] section, styles can be defined for menus in which all visual properties of a menu are defined. The individual properties can in turn be overwritten during the configuration of the respective menu.

Description of the menu style properties: See above.

## Example 4

```
[menustyles]
MENUSTYLE001= name:=msVerticalMenu, color:=59/76/88, layout:=column,
margin:=0, spacing:=1
MENUSTYLE023= name:=msGrid, color:=59/76/88, layout:=grid,
margin:=10, spacing:=1
```

Finally, the menu still needs to be assigned to a frame. The "menu" frame property is used for this:

```
[frames]
```

```
FRAME003= name:=frame3, x:=0, y:=970, width:=460, height:=110,
menu:=menu2
```

### 3.17.2.4 Menu actions

The actions to be performed when a menu item is pressed are defined in the onClicked property of the menu items. This property can include one or more command statements (separated by semicolons), which are executed one after the other.

The following commands are available:

| Command | Meaning |
|---|---|
| `showDisplay(displayname)` | Switches to the display named `displayname`<br>The target display must be configured in the same display configuration file. |
| `showApp(framename, appname)` | Displays application `appname` in the `framename` frame<br>If the application is already visible in another frame of the display, it is moved from there to the frame specified here. If possible, the freed frame is filled with the application displaced from `framename`. If this happens, it is referred to as application swapping. Which applications can be displayed in the released frame is specified in the "runableApps" property of this frame.<br>You can prefix the frame name with a display name (`displayname.framename`) and change the frame assignment in hidden displays.<br>Instead of `appname`, you can also specify `framename.currentApp` or `displayname.framename.currentApp`. `showApp()` then moves the displayed application in the `framename` or `displayname.framename` frame into the specified frame. |
| `showPopup(framename, appname)` | Displays application `appname` in the `framename` frame as popup window.<br>It is not permissible that frame `framename` belongs to a display, i.e. it is not specified in the `Frames` attribute for an existing display. It is automatically displayed with the `showPopup()` call and placed over the currently active display.<br>The only applications allowed are the applications created on the basis of the `SlGfwSideScreenDialog` class. SINUMERIK Operate and OEM frame applications cannot be displayed as popups. If you use an application that was developed based on the `SlGfwSideScreenDialog` class as a popup, you need to declare its window as a TopLevel window, which means the Qt::WindowStaysOnTopHint bit must be set in the `windowFlags()`.<br>Only one popup can be open at a time. |
| `hidePopup(appname)` | Hides an application displayed with `showPopup()` or `togglePopup()` |
| `togglePopup(framename, appname)` | Displays application `appname` in the `framename` frame as popup window. If the `appname` application is already displayed as popup window, then it is hidden again. |
| `showMenu(framename, menuname)` | Shows the `menuname` menu in the `framename` frame. |

| Command | Meaning |
|---|---|
| showPopupMenu(framename, menuname) | Shows the menu menuname in the framename frame as popup menu. |
| hidePopupMenu(menu name) | Hides a pop-up menu displayed with showPopupMenu() or togglePopupMenu(). |
| togglePopupMenu(framename, menuname) | Shows the menu menuname in the framename frame as popup menu. If menu menuname is already displayed as popup menu in the framename frame, then it is hidden again. |
| setMirrorMode(true\|false) | Enables (true) or disables (false) mirror mode<br>When mirror mode is enabled, the frames of all displays in which mirror mode has not been disabled with enableMirrorMode(false) are mirrored horizontally. In other words, frames that were previously displayed on the left are now displayed on the right and vice versa. The size of the frames does not change. |
| toggleMirrorMode() | Toggles the mirror mode<br>If it was previously enabled, it is now disabled and vice versa. |
| sendCmd(appname, cmd1, ...) | Sends a command to the appname application. If the target application is appname SINUMERIK Operate (appname="OPERATE"), the operating area of SINUMERIK Operate can be selected in this way,<br>e.g. sendCmd(OPERATE, AreaMachine)<br>The identifier for the operating area (AreaMachine) can be taken from the systemconfiguration.ini file (see [areas] section). If the target application is an application created on the basis of the SlGfwSideScreenDialog class, this application is given a command of the type SL_GFW_MSG_DISPCONFIG_CMD in its onMessage() implementation.<br>The commands cmd1,…, cmd10 are given as message data in the form of a string in which the individual parameters (cmd1 … cmd10) are separated by the "\|" symbol.<br>Commands to Windows applications (.EXE) are not possible with sendCmd()! |

## Examples

```
[menuitems]
MENUITEM001= name:=item1, onClicked:=
"showApp(frame1,OPERATE);showApp(frame2,myDialog)", text:="HMI"
```

Displays SINUMERIK Operate in "frame1" and the "myDialog" application in "frame2" on the current display.

```
MENUITEM003= name:=item3,
onClicked:="showApp(full.frame1,frame2.currentApp);showDisplay(full)
", text:="Full frame"
```

Sets the currently displayed application (currentApp) from "frame2" to "frame1" in the "full" display. Thereafter, the display is switched to "full".

```
MENUITEM004= name:=item4, onClicked:=
"sendCmd(OPERATE,'AreaMachine')", text:="Machine"
```

Selects the machine operating area in SINUMERIK Operate.

```
MENUITEM005= name:=item5,
onClicked:="sendCmd(myDialog,'doFun','42')", text:="Machine"
```

Sends message SL_GFW_MSG_DISPCONFIG_CMD with the data "doFun|42" to the "myDialog" application (requires the use of CreateMyHMI/3GL).

### 3.17.2.5 Applications

The applications used in a display configuration must be specified in the systemconfiguration.ini file.

There are two types of applications:

- Type A: Applications created on the basis of the SlGfwSideScreenDialog class
- Type B Standard Windows applications (.EXE)

All applications are automatically started by the Display Manager at system start and closed again when the system is shut down.

Type A applications are declared in the [dialogs] section of the systemconfiguration.ini file in the number range 500 - 999 (OEM area).

Type B applications are handled like OEM frame applications and are therefore declared in the [processes] section. In addition, the nPlacementMode parameter must be set to 3 for the applications in the C:\Program Files (x86)\Siemens\MotionControl\compat\user\OEMFRAME.INI file.

The following applications are included in the standard version:

- PDF Viewer
- VNC Viewer
- Virtual Keyboard
- Virtual MCP
- Application for displaying widgets
- SINUMERIK Operate

#### PDF Viewer

The configuration in the standard version:

```
[dialogs]
DLG107= name:=SlPdfApp,
implementation:=sldmpdfviewerapp.SlDmPdfViewerApp,
```

```
process:=SlHmiHost1, preload:=false cmdline:="-pdfFile
gestenbedienung.pdf -matteColor #3b4c58"
```

The application name to be used for configuration within a display configuration:
`SlPdfApp`

The document to be displayed is specified in the "cmdline" parameter. The two parameters have the following meaning:

- pdfFile: File to be displayed in the PDF Viewer. The file can be in the appl or hlp directories.
- matteColor: Background color of the PDF Viewer (color must be hex coded).

Additional instances of the PDF Viewer can be declared within a display configuration. The "name" parameter obtains the name of the additional instance and the "cmdline" parameter specifies the document to be displayed by this instance, for example:

```
DLGxxx= name:=anotherPdf,
implementation:=sldmpdfviewerapp.SlDmPdfViewerApp,
process:=SlHmiHost1, preload:=false cmdline:="-pdfFile
mydocument.pdf -matteColor #3b4c58"
```

## VNC Viewer

The configuration in the standard version:

```
[dialogs]
DLG109= name:=SlVncApp,
implementation:=sldmvncviewerapp.SlDmVncViewerApp,
process:=SlHmiHost1, preload:=false, cmdline:="-configuration MyPC"
```

The application name to be used for configuration within a display configuration:
`SlVncApp`

The VNC Viewer is configured in the `slvncconfig.ini` file. The name of the section containing the configuration for a specific VNC Viewer instance is specified in the "cmdline" parameter as the value of the "-configuration" argument - "MyPC" in the example above.

The following parameters must be specified for each VNC Viewer instance:

- host: IP address or DNS name of the host on which the associated VNC server is running.
- port: TCP port that the server uses.
- Password (optional): Password for protected access to the VNC server.

### Example

For configuration in the `slvncconfig.ini` file:

```
[MyPC]
host=172.218.152.43
port=5900
password=secret
```

## Virtual Keyboard

Configuration in the standard version:

```
[dialogs]
DLG112= name:=SlKeyboardApp,
implementation:=sldmvirtualkeyboardapp.SlDmVirtualKeyboardApp,
process:=SlHmiHost1, preload:=false, cmdline:="-settingsFile
sldm_keyboard.ini"
```

The application name to be used for configuration within a display configuration:

```
SlKeyboardApp
```

The application may only be used in one instance within a display configuration.

In the standard version, three sizes are pre-configured for the virtual keyboard (see file `sldm_1920x1080.ini`):

```
[760x505]
Num= KeyboardLayout:="numpadlayout", x:=430, y:=270
Alpha= KeyboardLayout:="nclayout", x:=92, y:=50
KeyHeight=47

[1024x242]
Num= KeyboardLayout:="numpadlayout", x:=784, y:=25
Alpha= KeyboardLayout:="nclayout", x:=206, y:=25
KeyHeight=47

[824x210]
Num= KeyboardLayout:="numpadlayout", x:=584, y:=22
Alpha= KeyboardLayout:="nclayout", x:=5, y:=22
KeyHeight=47
```

The section name corresponds to the respective frame size (specification in pixels). The attributes have the following meanings:

| Property | Description |
|---|---|
| `KeyboardLayout` | Layout of the virtual keyboard. Two layouts are available:<br>• `numpadlayout`: Only has keys for inputting numbers<br>• `nclayout`: Complete NC keyboard |
| `x, y` | Position of the upper left corner of the virtual keyboard in relation to the upper left corner of the frame in which the keyboard is displayed. |
| `KeyHeight` | Height (and width) of the keys on the virtual keyboard. |

If further frames with other sizes are configured in a display configuration in which the virtual keyboard is to be shown, the file `sldm_1920x1080.ini` must be supplemented with the corresponding sections and stored in the oem directory.

## Virtual MCP

Configuration in the standard version:

```
[dialogs]
DLG111= name:=SlMcpApp,
implementation:=sldmsidescreenapp.SlSideScreenDialog,
process:=SlHmiHost1, preload:=false, cmdline:="-sidescreen1
sldmmcppage.ini"
```

The application name to be used for configuration within a display configuration:
`SlMcpApp`

The application may only be used in one instance within a display configuration.

### Application for displaying widgets

Configuration in the standard version:

```
[dialogs]
DLG110= name:=SlWidgetsApp,
implementation:=sldmsidescreenapp.SlSideScreenDialog,
process:=SlHmiHost1, preload:=false, cmdline:="-sidescreen1
sldmwidgets1.ini -sidescreen2 sldmwidgets2.ini -spacing 3"
```

The application name to be used for configuration within a display configuration:
`SlWidgetsApp`

The application may only be used in one instance within a display configuration.

### SINUEMRIK Operate

The application name to be used for configuration within a display configuration:
`OPERATE`

SINUMERIK Operate may only be used in one instance within a display configuration.

It should also be noted that SINUMERIK Operate can only be displayed/executed in certain sizes. The currently supported sizes (in pixels) are:

- 640x480
- 760x505
- 800x480
- 800x600
- 1024x768
- 1024x1014
- 1080x810
- 1280x768

- 1280x800
- 1280x1024
- 1300x768
- 1366x768
- 1440x1080
- 1454x1080
- 1854x1014
- 1854x1080
- 1920x1080

### Example of Type B applications

Configuration in the `systemconfiguration.ini` file with the `notepad.exe` application as an example:

```
[processes]
PROC500=process:=Notepad, image:=notepad, cmdline:="C:\Windows
\system32\notepad.exe", oemframe:=true, windowname:="Unknown -
Editor", classname:="Notepad", processaffinitymask:=0xFFFFFFFF,
deferred:=true
```

The application name to be used for configuration within a display configuration:

```
Notepad
```

The application may only be used in one instance within a display configuration.

Configuration in the `oemframe.ini` file with the `notepad.exe` application as an example:

```
[notepad]
nPlacementMode=3
WindowStyle_Off = 15597568
```

Type B applications may only be executed in one instance.

### 3.17.2.6 Global settings

In addition to the configuration of the displays and their layouts, there is also a `[miscellaneous]` section. Settings can be made there that apply to all displays of a display configuration (configuration file).

| Entry | Meaning |
|---|---|
| `startupdisplay` | Name of the display that is to be selected after starting the Display Manager. If this information is omitted, the display configured first is used as the `startupdisplay`. |
| `hmihostofmenus` | Specifies the host process in which the menus associated with a display configuration are executed. Default is `SlHmiHost1`. |
| `backgroundcolor` | Background color of the desktop, which can be visible in the gaps between the frames depending on the configuration of the frames. |
| `textfile` | Language-dependent texts can be used in the same way as with SINUMERIK Operate. One or more text files can be specified. |
| `textcontext` | In addition to specifying the text context for configuration of the menus or the menu items, a global text context can be specified here, which then applies to the language-dependent texts of all menu items. |

## Example

```
[miscellaneous]
textfile=sometextfile, anothertextfile
textcontext=MyGlobalTextContext
```

#### Color information

Colors can be specified via decimal RGB values, for example in the form:
`color:=255/255/255` or `color:="255, 255, 255"` or `color:=255 255 255`

**Note**

Note the required quotes when commas are used as delimiters, because commas are also the separator between the property information.

Alternatively, the usual hexadecimal notation can be used with HTML, for example:
`color:=#FFFFFF`

Or the "X11 Named Colors" can be used that offer Qt as native:
`color:=ForestGreen` or `color:=Forest Green`

# Licensing

# 4

Licenses are required for the activated options. When you purchase licenses you receive a "license key" that contains all options requiring a license and which is only valid for your CompactFlash card.

## Requirement

If you wish to set or reset the options for a machine, you need at least access level 3 (user).

You can only license options for a machine from this access level and higher.

## General information

| | |
|---|---|
| Serial number of the CompactFlash card | Permanently defined serial number of the CompactFlash card |
| Hardware type | Data of the control |
| Machine name/No. | Displays the data saved in machine data MD17400 $MN_OEM_GLOBAL_INFO.<br>Only with access level 1 (manufacturer) can a display field be edited. |
| License key | Information about options that require a license |

## Procedure




1. Select the "Setup" operating area.



2. Press the menu forward key.







3. Press the "Licenses" softkey.
   The "Licensing" window opens.
   The general data for control and licenses is displayed.

## License database

You purchase licenses via the Web License Manager in the Internet, administered by Siemens Industry.

### References

For a detailed description, please refer to the following documentation:

CNC Commissioning Manual: NCK, PLC, drive; Chapter "Licensing"

### Deactivate options

When importing old archives from other HMI systems, it can occur that options are set which are no longer required in the actual system – however, these are subject to the license check:

| Order number | Option |
|---|---|
| 6FC5800-0AP04-0YB0 | Machining step programming |
| 6FC5800-0AP11-0YB0 | Manual machine |
| 6FC5800-0AP15-0YB0 | ShopTurn HMI/ShopMill HMI for SINUMERIK 840Di sl incl. HMI-Advanced |
| 6FC5800-0AP20-0YB0 | 3D simulation of the finished part |
| 6FC5800-0AP21-0YB0 | Simulation, milling (2D dynamic; 3D static) |
| 6FC5800-0AP23-0YB0 | Simultaneous recording, ShopMill (real-time simulation) |
| 6FC5800-0AP24-0YB0 | Simultaneous recording, ShopTurn (real-time simulation) |

## 4.1 Determining the license requirement

### Procedure




1. Press the "All options" softkey to list all the options that can be selected for this controller.

   - OR -




   Press the "Missing lic./opt." softkey.

   You receive an overview of all options that

   - are not adequately licensed
   - are licensed, but which have not been set

2. Activate or deactivate the required options in the "Set" column:
   - Activate the checkbox or
   - Enter the number of options.

   Options shown in red are activated, however, not yet licensed or not adequately licensed.

   - OR -




   Press the "Set option acc. to license" softkey to activate all of the options contained in the license key.

   You obtain a confirmation prompt that you must confirm with "OK".




3. To activate new selected options, press the "Reset (po)" softkey. A safety prompt appears.

   For several options, you will have to also restart SINUMERIK Operate. Corresponding prompts will appear in the user response line.




4. Press the "OK" softkey to trigger a warm restart.

   - OR -




   Press the "Cancel" softkey to cancel the process.

**Note**

If you activate the "Set option acc. to license" function, the sum of the axes that can be activated may be exceeded. Therefore, reduce the number of actually used axes in the corresponding options.

## Searching for options

You have the possibility of finding options using the search function.

1. The window "Licensing: All Options" or "Licensing: Non-licensed Options" is open.




2. Press the "Search" softkey and enter the required option in the search form.

   The cursor is positioned on the first entry that corresponds to the search term.




3. Press the "Continue search" softkey if the term that was found is not the option that you were looking for.

## Additional search option




Press the "Go to beginning" softkey to start the search at the first option.




Press the "Go to end" softkey to start the search at the last option.

**Note**

**Use of non-licensed options**

Before licensing options, you can also activate them temporarily without the license keys and use them for test purposes. In this case, the control displays alarm 8080/8081 periodically, indicating that the option used has not yet been assigned the required license. Further, in this state, the NC cannot be started.

## 4.2 Activating the test license

If you wish to use options without any restrictions – and without purchasing a license – you can activate these for a limited time using the test license. Six test license periods are available. The first test license period has a longer operating time. The operating time depends on the NCU being used. Generally, the additional test license periods are 150 operating hours of the control system.

**Note**

A test license supports "short-term use" of the software in a non-productive context, e.g. use for testing and evaluation purposes; it can be transitioned into another license.

### Duration of the first test license period

| NCU | Operating time duration |
|---|---|
| 710.3 | 1000 h |
| 720.3 | 2000 h |
| 730.3 | 3000 h |

### Requirement

If you wish to activate the test license, you require access level 2 (service).

### Procedure




1. Press the "Activate test license" softkey.
   A message informs you of the remaining number of test license periods.




2. Press the "OK" softkey.
   The test license period is now active.
   The active test license periods and the remaining operating time are now displayed.

**Note**

In the message line, the control displays the operating time still available for the current test license period every hour.

**Note**

**Interruption of the test license**

A test license period is interrupted in the following cases:

- If no axis is operated actively, e.g. during simulation mode.
- If a new valid license key is entered.
- If all non-licensed options are reset.
  The test license continues when a non-licensed option is set again.

**Note**

**End of the test license**

A test license period ends when the time has expired.

No further test license can be activated on the NCU if all test license periods have expired.

## 4.3 Exporting license requirements

After you have determined the license requirement, you can export the data via the set-up external storage locations.

### Procedure




1. Press the "Export license requirement" softkey.
   A window opens listing the storage locations that have been set up.
2. Select "USB" for example, if you export the licenses on a USB flash drive.




3. Press the "OK" softkey.
   The set options, the options which are not set but are licensed options, the license requirement and the license key are stored in an Alm file.
   In addition, an Internet connection is established for the Web License Manager.

### References

A detailed description of the license management is provided in the following documentation:

CNC Commissioning Manual: NCK, PLC, drive; Chapter "Licensing"

## 4.4 Reading in the license key

You have the following options of communicating the license key to the control:

- Enter manually
- Read in electronically

### Procedure

Overview — The "Licensing" window is open.

**Enter license key manually**

1. If you obtained the license key via the Web License Manager, enter the license key manually in the field "You can enter a new license key here".
2. Press the <INPUT> key.
   If the license key is valid, the message "License key set" is displayed.

**Read in license key electronically**

Read in license key — If you have an alm file with an appropriate license key, then press the "Read-in license key" softkey.

Open the appropriate storage location (e.g. USB) with "OK".

...

OK — The matching .alm file is detected automatically and the license key is read in.

Note:

If several matching .alm files are in the directory, they will be offered to you in a subsequent selection dialog box.

Select the desired file.

## 4.5 Backing up the license of a new CompactFlash card.

If, for example, you want to replace a defective card on the control and transfer an existing archive to a new empty CompactFlash card, you must back up the license of the new CF card before restoring. If you restore without backing up, the license key of the CompactFlash card will be overwritten.

### Requirement

To back up the license key, you require a multi-card reader with USB on a PG/PC.

### Procedure

1. The CompactFlash card has been inserted into the multi-card reader, which is connected to the PG/PC.
2. Open directory cards/keys/sinumerik/.
   Here you will find the license key in the form of a txt file.
3. Copy the license key to your back-up directory.
4. Insert the CompactFlash Card into the control and perform the restore action.
5. Insert the CompactFlash card into the multi-card reader again.

6. Copy the license key, renamed keys.txt, back to directory cards/keys/sinumerik.
7. Insert the CompactFlash card into the control again.
   After the control has been powered up again, the license key will be installed from the file keys.txt onto the system and be activated. The system renames the txt file to the installation date.

### What to do if you lose the license key

To obtain a valid license key for the new, empty CompactFlash Card, please contact the SINUMERIK hotline. They will then send you a new license key without delay.

# Creating drive connections 5

Up to 21 connections to so-called logical drives (data carriers) can be configured. These drives can be accessed in the "Program manager" and "Startup" operating areas.

The following logical drives can be set up:

- USB interface
- CompactFlash card of the NCU, only for SINUMERIK Operate in the NCU
- Network drives
- Local hard disk of the PCU, only for SINUMERIK Operate on the PCU



**Software option**

In order to use the CompactFlash card as data carrier, you require the "Additional HMI user memory on CF card of the NCU" option (not for SINUMERIK Operate on PCU or PC).

**Note**

The USB interfaces of the NCU are not available for SINUMERIK Operate and therefore cannot be configured.

## 5.1 Setting up drives

The "Set up drives" window is available in the "Start-up" operating area for configuring the softkeys in the Program Manager.

Information on the procedure and configuration can be found in the following documentation:

### References

Operating Manual, Universal/Turning/Milling/Grinding, Chapter "Managing programs" > "Setting up drives"

### File

The created configuration data is stored in the "logdrive.ini" file. This file is located in the /user/sinumerik/hmi/cfg directory.

### General information

**Note**

We do not recommend that a USB front interface is permitted for part program execution using EES if a system has several local USB front interfaces (several TCUs, the use of a HT8, etc.). The connection to the USB FlashDrive can be interrupted as a result of the TCU switchover. This setting should be made using the checkbox "Use this drive for program execution with EES".

**Note**

If a global USB front interface should be used via PCU, then the PCU must be configured as server via the VNC connection, see Chapter "Configuring VNC connections (Page 225)".

## 5.2 Setting EXTCALL

Using the EXTCALL command, a program on an external program memory can be called from a part program.

EXTCALL must be switched off, when in certain configurations two SINUMERIK Operate are connected to one NCU.

### Activating the function

Processing EXTCALL calls is switched-on or switched-off using the following display machine data.

| MD9106 $MM_SERVE_EXTCALL_PROGRAMS | |
|---|---|
| = 0 | HMI ignores EXTCALL instructions and selection via PLC |
| = 1 | HMI processes EXTCALL instructions and selection via PLC (default value) |
| = 2 | HMI processes EXTCALL instructions and ignores selection via PLC |
| = 3 | HMI ignores EXTCALL instructions and processes selection via PLC |

**Note**

A main program selected from an external program memory is selected again automatically after a power-on if the same program memory is still available and execution of EXTCALL calls has been activated in MD9106.

## 5.3 Creating a global part program memory

With the global part program memory (GDIR), a structure is provided analog to the passive file system of the NC. It is integrated in the search path of the NC for subprogram calls. If you create a GDIR, all participants in the system have access to the same logical drive. The participants can directly execute part programs from the common program memory. The GDIR so replaces or extends the part program memory of the NC.

**Note**

The global part program memory plays a role only for the EES function. The creation of a GDIR is, however, not essential for the EES operation.

### Possible procedure

1. Declare the global part program memory in the "Creating drives" window
2. Shift the content of the local program memory with SPF, MPF and WCS directory to an external drive, e.g. to a USB-FlashDrive.
3. Copy the directories that have been backed up to the global part program memory.

Additional information about managing the program memory is available in the following documentation:

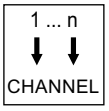
### References

Job Planning Programming Manual, Chapter "File and Program Administration" > "Program memory"

# Channel menu

# 6

## 6.1 Configuring a channel menu




The channel menu is activated for display via the channel switchover key.

The channel menu is used to switchover the NC channel displayed in the SINUMERIK Operate or more precisely at the operator station. If a channel menu has been configured, when the channel switchover key is pressed, the channel menu for the channel switchover is displayed to make the appropriate operator action. If there is no channel menu, then the channel switchover key results in the HMI display advancing to the next NC channel.

### Configuring

The configuration is made in the "netnames.ini" file for the operating software, as well as in the particular "config.ini" files for the individual operator stations.

The "netnames.ini" file must be generated using an editor.

### Storage

The "netnames.ini" file is stored for SINUMERIK Operate in the following directory:
/oem/sinumerik/hmi/cfg or /user/sinumerik/hmi/cfg

The "config.ini" file is located in the relevant directory:

- On the NCU:
  /user/common/tcu/<*TCU name*>/common/tcu/config.ini
- On the PCU:
  F:\user_base\common\tcu\<*TCU name*>\common/tcu/config.ini

For <*TCU name*>, the TCU name of the operator station should be used that was assigned at the system when a TCU powered-up for the first time.

### Using the channel menu

- For a system comprising several operator stations and several NCUs within the framework of operator unit management, if the NC channels of several NCKs are operated from one operator station.
- For a simple system with many NC channels, as an alternative to the standard mechanism, the channel switchover can be made using the channel switchover key.

The basic philosophy of the channel menu is to provide - in a selection menu - the operator with a view of the NC channels of all NCUs that are to be operated with the channel menu. In conjunction with this, a logical grouping/structuring is introduced using the so-called channel groups. The desired NC channel can be directly selected using the selection menu. Without

a channel menu, when pressing the channel switchover key, the NC channels must be scrolled through consecutively until the desired NC channel has been selected. An operator station can only be switched over across NCUs using the channel menu.

## 6.2 Constraints

- Only one SINUMERIK Operate on a PCU/ PC may be permanently connected to one NCK. It is not possible to switch over to another NCK.
- SINUMERIK Operate of an NCU can only operate with its own NCK and cannot be switched over to another NCU.

## 6.3 General structure of the channel menu

### Overview

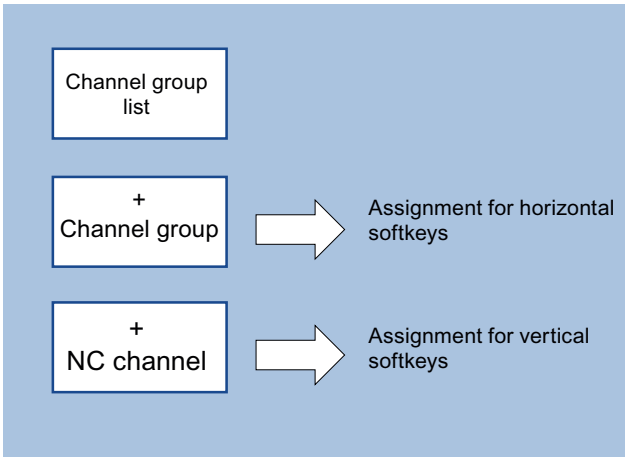
For the channel menu, a logical view is defined across all of the channels of all NCKs involved; this logical view is structured in channel groups and this in turn in specific NC channels.

A channel group list is defined for the channel menu in order to be able to switch over operation. A channel group list comprises one or several channel groups. In turn, a channel group comprises one or several NC channels.

In the channel menu, the channel groups can be selected using the horizontal softkeys.

The channels of a selected channel group can be selected with the vertical softkeys.

A certain NC channel of a specific NCU is always a switchover target that can be selected. A switchover to another channel triggered via the channel menu can implicitly mean switchover to another NCU.



**Note**

In a system, for example, a channel group can correspond to a machining station, the channels contained in it, machining units.

Up to 32 channel groups each with up to 8 channels can be configured in a channel menu.

### Configuring the total structure

For a system, the total structure of the channel menu is configured in the "netnames.ini" file. The configuring in the "netnames.ini" file includes all switchover targets that are used at the system. This configuration must be saved or distributed in a standard form across all SINUMERIK Operate systems that provide the channel menu.

An individual channel menu for the operating station is configured for each operating station in the particular "config.ini" file.

### Two-stage procedure

1. In the first stage, all channel groups that are used in the system are defined in the "netnames.ini" file, irrespective of the actual operating stations.
2. In the second stage, the version of the channel menu for the specific operating station is defined in the "config.ini" files belonging to the station.

## 6.4 Defining channel groups in the "netnames.ini" file

### Defining channel groups

All channel groups that are used at the system are defined in the "netnames.ini" file. Independent of the specific operator stations.

A separate section [CH_GRP_x] (1 <= x <= 32) should be created for every channel group.

The entries required for a channel group are as follows:

- The text to be displayed in the channel for the channel group at the corresponding horizontal softkey.
- The list of jump destinations summarized in this channel group, i.e. NC channels. The jump destinations are defined in a consecutively numbered list with the following entries:
  - Member*y*.Target: Jump destination
  - Member*y*.Text: Softkey text to be displayed in the channel menu (1 <= y <= 8).

### "netnames.ini" file

A channel group, e.g. channel group 1 is therefore specified as follows:

```
[CH_GRP_1]                    ;  Name of the channel group; the notation
                              ;  must follow this scheme. The groups are
                              ;  consecutively numbered at the last
                              ;  position.
Text = Process1               ;  Softkey text for the channel group for
                              ;  display in the channel menu (horizontal
                              ;  softkey) switchover designation 1 of the
                              ;  channel group.
Member1.Target =              ;  Computer name (host name) or the IP
[<HMI>.]<NC>.#<channel        ;  address of the computer on which the HMI
number>                       ;  to be selected runs (NCU or PCU):
                              ;  <HMI> = <IP address or name of the HMI
                              ;  host (PC/PCU)>
                              ;  <HMI> can be omitted if the HMI runs
                              ;  locally on the NCU
                              ;  <NC> = <IP address or name of the NC>
                              ;  Alternatively, the machine name can also
                              ;  be specified from the mmc.ini.
                              ;  Using ".#Channel number", optionally the
                              ;  NC channel can be specified that should
                              ;  be taken when switching over.
                              ;
Member<No>.Text = NCU730_K1   ;  Softkey text for the display in the
                              ;  channel menu (vertical softkey). If, for
                              ;  "Target" a channel number is specified,
                              ;  then instead of a text, using the fixed
                              ;  value "$MC_CHAN_NAME", it can also be
                              ;  defined that the text to be displayed
                              ;  should be the channel name defined in the
                              ;  NCK machine data.
                              ;
Member2.Target =
Member2.Text                  ;  Same as for the vertical softkey 2
```

## 6.5 Configuring operating stations in the "config.ini" file

The version of the channel menu is individually defined for the operator stations in the "config.ini" files belonging to the station. The data defined in the "netnames.ini" file is used. Here, the channel groups are assigned a softkey in the channel menu.

### "config.ini" file

- The data is defined in section [T2M2N].
- The references CH_GRP_xx of the softkeys refer to the channel groups which are defined in the "netnames.ini" file. This notation is binding.
- SKx in this case, designates the xth horizontal softkey in the channel menu.

### Example

```
...
[T2M2N]
SK1=CH_GRP_1
SK2=CH_GRP_3
SK8=CH_GRP_15
```

## 6.6 Example: Settings for 1:N

### 1:N configuration

For a 1:N configuration the following requirements apply:

- A PCU with SINUMERIK Operate can be connected to a maximum of four NCUs.
- SINUMERIK Operate must be switched off on all NCUs.
- The following configuration files must be adapted:
  siemens/sinumerik/hmi/template/cfg/**netnames.ini**
  siemens/sinumerik/hmi/template/cfg/**mmc.ini**

This is illustrated in the following sections using an example with 1 PCU and 2 NCUs.

### Example mmc.ini

```
[Global]
NcddeMachineNames=net, NCU840D_1, NCU840D_2
NcddeDefaultMachineName=NCU840D_1

[NCU840D_1]
ADDRESS0=192.168.214.1,LINE=14,NAME=/NC,SAP=040d,PROFILE=CLT1__CP_L4_INT
ADDRESS1=192.168.214.1,LINE=14,NAME=/PLC,SAP=0202,PROFILE=CLT1__CP_L4_INT
ADDRESS2=192.168.214.1,LINE=14,NAME=/CP,SAP=0502,PROFILE=CLT1__CP_L4_INT
ADDRESS10=192.168.214.1,LINE=14,NAME=/DRIVE_00_000,SAP=0201,
SUBNET=0046-00000000:000,PROFILE=CLT1__CP_L4_INT
ADDRESS11=192.168.214.1,LINE=14,NAME=/DRIVE_03_003,SAP=0900,
PROFILE=CLT1__CP_L4_INT
```

```
[NCU840D_2]
ADDRESS0=192.168.214.2,LINE=14,NAME=/NC,SAP=040d,
PROFILE=CLT1__CP_L4_INT
ADDRESS1=192.168.214.2,LINE=14,NAME=/PLC,SAP=0202,
PROFILE=CLT1__CP_L4_INT
ADDRESS2=192.168.214.2,LINE=14,NAME=/CP,SAP=0502,
PROFILE=CLT1__CP_L4_INT
ADDRESS10=192.168.214.2,LINE=14,NAME=/DRIVE_00_000,SAP=0201,
SUBNET=0000-00000000:000,PROFILE=CLT1__CP_L4_INT
ADDRESS11=192.168.214.2,LINE=14,NAME=/DRIVE_03_003,SAP=0900,
PROFILE=CLT1__CP_L4_INT
```

## Example netnames.ini

```
;*****************************************************************
;     netnames.ini      - EXAMPLE -
;
; This is an example of a netnames.ini for 1:N NCU-switching in
; a configuration of 3 NCU and a PCU50.
;*****************************************************************
[own]
owner = HMI

;**************************
; ChanMenu
;**************************
[chan HMI]
ShowChanMenu = true

;*****************************************************************
;Channel-groups
;Correlation from channel-group to horizontal softkeys
; -> see config.ini of the TCU
;The section [T2M2N] can also be created in the netnames.ini file
;instead of config.ini file. This configuration applies to all TCU,
;which are not configured by their config.ini.
;*****************************************************************
[T2M2N]
SK1 = CH_GRP_1
SK2 = CH_GRP_2

[CH_GRP_1]

Text = 1. NCU
Member1.Target = 192.168.214.241.192.168.214.1.#1
Member1.Text = NCU1%nKanal 1

Member2.Target = 192.168.214.241.192.168.214.1.#2
Member2.Text = NCU1%nKanal 2

[CH_GRP_2]
```

```
Text = 2. NCU
Member1.Target = 192.168.214.241.192.168.214.2.#1
Member1.Text = NCU2%nKanal 1

Member2.Target = 192.168.214.241.192.168.214.2.#2
Member2.Text = NCU2%nKanal 2
```

**Note**

For switching over an MCP with the operating software, the PLC block FB9 must not be used (FB9 must not be active). If necessary, the MCP must be switched over in the PLC user program.

### Solution

In the OB100 (call for FB1), set MCP1BusAdr to 255 and MCP1Stop to "true".

In the OB1, scan the "MMCBTSSready" (DB10.DBX108.3) signal. If this is set, write the MCP address from DB19.DBB123 (MCP index) to the DB7.DBB27 before removing the MCP stop bit DB7.DBX62.1. The MCP is now active.

If the "MMCBTSSready" (DB10.DBX108.3) is removed again (operating software switches to another NCU), the MCP stop bit DB7.DBX62.1 will have to be reset.

## 6.7 Distribution via job lists in the case of 1:N

### Simplified selection of programs

The function is only available in a group of several NCUs ands must be enabled in the "systemconfiguration.ini " configuration file:

```
[miscellaneous]

distributeJoblist=true
```

The sources are located on the local drive of the PCU or the PC.

The distribution of the programs is done by selecting a job list which is located in a workpiece. The job list may contain COPY and SELECT instructions:

```
; Copy A.MPF to NCU1:

COPY A.MPF NC=NCU1

; Copy B.MPF to NCU2:

COPY B.MPF NC=NCU2

; Copy all subprograms to all NCUs:

COPY *.SPF *

; Select A.MPF on NCU1 in channel 1.

SELECT A.MPF ch=NCU1,1
```

```
; Select B.MPF on NCU2 in channel 2.
SELECT B.MPF ch=NCU2,2
```

**Sequence:**

1. If a distributed job list is active during the selection, all distributed programs on the NCU are deleted without confirmation. If the workpiece on the NCUs is empty, it will be deleted there.
2. The programs are copied to the corresponding workpieces of the relevant NCU.
3. Existing programs are overwritten without any message.
4. Following a change in a distributed program on the local drive, the modified programs using the Editor are copied to the corresponding NCU.

## Sample structure

| /user/sinumerik/data/ prog/ JOBA.WPD | NCU1 | NCU2 |
|---|---|---|
| JOBA.JOB | WKS.DIR | WKS.DIR |
| A.MPF | WKS.DIR/JOBA.WPD | WKS.DIR/JOBA.WPD |
| B.MPF | WKS.DIR/JOBA.WPD/B.MPF | WKS.DIR/JOBA.WPD/B.MPF |
| UP1.SPF | JOBA.WPD/UP1.SPF | JOBA.WPD/UP1.SPF |
| UP2.SPF | WKS.DIR/JOBA.WPD/UP2.SPF | WKS.DIR/JOBA.WPD/UP2.SPF |

| /user/sinumerik/data/ prog/ wks.dir/JOBA.WPD | NCU1 | NCU2 |
|---|---|---|
| JOBA.JOB | WKS.DIR | WKS.DIR |
| A.MPF | WKS.DIR/JOBA.WPD | WKS.DIR/JOBA.WPD |
| B.MPF | WKS.DIR/JOBA.WPD/B.MPF | WKS.DIR/JOBA.WPD/B.MPF |
| UP1.SPF | JOBA.WPD/UP1.SPF | JOBA.WPD/UP1.SPF |
| UP2.SPF | WKS.DIR/JOBA.WPD/UP2.SPF | WKS.DIR/JOBA.WPD/UP2.SPF |

## Supplementary conditions

Special behavior:

- Distributed programs are retained during a restart of the operating software.
- On the local drive, distributed programs can neither be deleted nor renamed nor overwritten using copy/move.
- Changes of distributed programs on the NCU are lost, if they are not manually saved.
- The program correction in the machine is not offered.
- When creating a commissioning archive, all data of the local drive is also automatically saved.

Restrictions:

- In the event of changes of distributed programs on the local drive using the Editor, it should be noted that only the channel states are monitored on the NCU currently connected. If the program is still interrupted or active on another NCU, it cannot be correctly distributed after the change.
- The display of the active block, the marking of the block search as well as the read-only handling of the program opened in the Editor are executed only for the active programs of the currently connected NCU.
- In the Editor, programs are always opened in the context of the currently connected NCU. This can lead to a faulty step recognition with distributed programs.
- In the current block display in the "Machine" operating area, you can display only the active/ distributed programs of the currently connected NCU.
- The simulation of job lists via several machines is not possible.
- If `distributeJoblist=true` is configured in the systemconfiguration.ini file, then part or subprograms can no longer be selected via the "Program Manager" operating area.

# Machine and setting data

# 7

## 7.1 Machine and setting data

### Overview

| Prefix | Number ranges | Machine/setting data | Storage in range (softkey) |
|---|---|---|---|
| $MM | 9000 - 9999 | Display machine data | Display MD |
| $MNS | 51000 - 51299 | General configuration machine data | General MD |
| | 51300 - 51999 | General cycle machine data | |
| $MCS | 52000 - 52299 | Channel-specific configuration machine data | Channel MD |
| | 52300 - 52999 | Channel-specific cycle machine data | |
| $MAS | 53000 - 53299 | Axis-specific configuration machine data | Axis MD |
| | 53300 - 53999 | Axis-specific cycle machine data | |
| $SNS | 54000 - 54299 | General configuration setting data | General SD |
| | 54300 - 54999 | General cycle setting data | |
| $SCS | 55000 - 55299 | Channel-specific configuration setting data | Channel SD |
| | 55300 - 55999 | Channel-specific cycle setting data | |
| $SAS | 56000 - 56299 | Axis-specific configuration setting data | Axis SD |
| | 56300 - 56999 | Axis-specific cycle setting data | |

### References

You can find a detailed description of the machine and setting data in the following documentation:

Machine Data and Parameters List Manual

## 7.2 Displaying/editing machine data

You can access the following machine data in the "Start-up" operating area.

- General machine data ($MN) and general configuration machine data ($MNS)
- Channel-specific machine data ($MC) and channel-specific configuration machine data ($MCS)
- Axis-specific machine data ($MA) and axis-specific configuration machine data ($MAS)
- Drive-specific machine data ($M_)
- Machine data control unit: Drive parameters
- Machine data feed-in: Drive parameters

- I/O component parameters: Drive parameters
- Communication parameters (CULINK, DMC): Drive parameters

**Note**

The display of the machine data / drive parameters depends on the configuration of your control.

Access authorization to the machine data operating area can be controlled by keyswitch position or password.

Read-access to machine data is possible as of access level 4 (keyswitch 3).

Machine data can be changed with access level 1 (manufacturer).

**⚠ WARNING**

**Incorrect parameterization**

Changes in the machine data have a considerable influence on the machine. Incorrect configuration of the parameters can endanger human life and cause damage to the machine.

## Information about machine data

The following information is displayed from left to right:

- Machine data number, with array index if applicable
- Machine data name
- Machine data value
- Machine data unit
- Effective

**Note**

If the machine data does not use units, the corresponding column is empty. If the data is not available, the hash symbol "#" is displayed instead of the value. If the value ends in an "H", it is a hexadecimal value.

The physical units of machine data are displayed on the right-hand side of the input field.

For each machine data item, an activation type can be read in the column on the right.

| | |
|---|---|
| so | Immediately active: No action required |
| cf | Configuration: "Set MD to active (cf)" softkey |
| re | Reset: <Reset> key on the machine control panel |
| po | POWER ON: "Reset(po)" softkey |

### Additional Information

In the footer you are given a brief explanation of the selected machine or setting data item.

## Procedure




1. Select the "Start-up" operating area.




2. Press the "Mach. data" softkey. Additional softkeys for the individual machine data areas are displayed.










3. Press the "General MD", "Channel MD", or "Axis MD" softkey.
   The window opens and the selected machine data is displayed.
4. Place the cursor on a machine data item.
5. Position the cursor on the entry to be modified and enter the required value.
6. Depending on the activation type, there are two ways of activating the settings:




   Press the "Activate MD (cf)" softkey. The value is applied.

   - OR -




   Press the "Reset (po)" softkey.
   A safety prompt appears.




7. Press the "OK" softkey to trigger a warm restart.

   - OR -




   Press the "Cancel" softkey if you do not want to apply the settings.

## Axis selection

If several components are available for the area selected (e.g. "Axis MD"), you can choose from the following options:







1. Press the "Axis +" or "Axis -" softkey.
   The values of the next (+) and the previous (-) axes are displayed.

   - OR -




   Press the "Direct selection..." softkey.
   The "Axis Direct Selection" window opens.
2. Select the required axis directly from those available in the drop-down list box.




3. Press the "OK" softkey to save the setting.

- OR -




Press the "Cancel" softkey to reject the settings.

## 7.3 Displaying/editing display machine data

You can access the display machine data via the user interface in the "Start-up" operating area.

### Procedure




1. Select the "Start-up" operating area.




2. Press the "Mach. data" softkey.



3. Press the menu forward key.
   Further softkeys are displayed.




4. Press the "Display MD" softkey.
   The display machine data appears.
5. Position the cursor on the entry to be modified and enter the required value.




6. Press the "Reset (po)" softkey.
   A safety prompt appears.




7. Press the "OK" softkey to trigger a warm restart.

- OR -




Press the "Cancel" softkey if you do not want to apply the settings.

## 7.4 Displaying/editing setting data

You can access the following setting data via the user interface in the "Startup" operating area.

- General setting data
- Channel-specific setting data
- Axis-specific setting data

## Procedure




1. Select the "Start-up" operating area.




2. Press the "Mach. data" softkey.



3. Press the menu forward key.
   The softkeys "General SD", "Channel SD", and "Axis SD" are displayed.










4. Press the relevant softkey to display a particular setting data area.
   The window opens and the selected setting data is displayed.

5. Place the cursor on the desired setting data and position the cursor on the entry to be changed. Enter the desired value.

6. Depending on the activation type, there are two ways of activating the settings:




   Press the "Activate MD (cf)" softkey.

   - OR -




   Press the "Reset (po)" softkey.
   A safety prompt appears.

7. Press the "OK" softkey to trigger a warm restart.




   - OR -




   Press the "Cancel" softkey if you do not want to apply the settings.

### Axis/channel selection

If several components are available for the area selected (e.g. "Axis SD" or "Channel SD"), you can choose from the following options:













1. Press the "Axis +" or "Axis -" or "Channel +" or "Channel -" softkey.
   The values of the next (+) and the previous axis/channel (-) are displayed.

- OR -




Press the "Direct selection..." softkey.
The "Direct Selection" window opens.

2. Select the required axis directly from those available in the drop-down list box.




3. Press the "OK" softkey to save the setting.

- OR -




Press the "Cancel" softkey to reject the setting.

## 7.5 Displaying/editing drive parameters

The "Control Unit MD", "Infeed MD" and "Drive MD" softkeys as well as the "I/O comp. MD" and "Comm. MD" softkeys on the menu forward key are available to display and edit drive parameters.

A separate list display is provided for each of these areas in which you can view and edit drive data.

### Precondition

Settings for drive parameters are protected with access level 2 (service).
The lists of the drive parameters are visible with access level 4 (keyswitch 3).

Please refer to the following documentation for information on commissioning the drive:

### References

CNC Commissioning Manual: NCK, PLC, drives

## Procedure




1. Select the "Start-up" operating area.




2. Press the "Mach. data" softkey.
   The "Control unit parameters", "Supply MD" and "Drive MD" softkeys are displayed.
3. Press the softkey of an area to show its parameters.
4. Place the cursor on the desired parameter.
5. Position the cursor on the entry to be modified and enter the required value.




6. In order to activate the settings, you have the option of saving or carrying out a reset.
   Press the "Save/reset" softkey.




7. Press the "Save" softkey.
   You are prompted "What do you wish to save?"
   You can make the following selection per softkey:
   - Actual drive object
   - Drive unit
   - Drive system

   - OR -




Press the "Reset (po)" softkey.
You are prompted as to whether you wish to initiate a warm restart for the NCK and the complete drive system.




Confirm the prompt with the "Yes" softkey.




Reject the warm restart with the "No" softkey.

## Extended reset

If you wish to carry out a warm restart only for specific objects or specific areas, then additional softkeys are available for this purpose.




Press the "Extended reset" softkey.
You are prompted to enter for which objects you wish to initiate a warm restart.
You can make a selection with the following softkeys:




Only the NCK is restarted.

| | |
|---|---|
| NCK and drive sys. | All of the control units and the NCK in the system are restarted. This softkey corresponds to the "Reset (po)" softkey if you confirm the prompt with "Yes". |
| NCK, HMI, drive sys. | All of the control units and the NCK in the system as well as the operator software are restarted. |

### Hiding the expanded softkey bar

The "Extended reset" softkey is defined in the "slsuconfig.ini" file. If the softkey is not to be displayed, proceed as follows:

1. Copy the "slsuconfig.ini" file from the /**siemens**/sinumerik/hmi/template/cfg/ directory.
2. Insert the file into the following directory:
   /**user**/sinumerik/hmi/cfg/
3. Open the file and in the section
   [Softkeys] and enter
   AdvancedReset = **0**

## 7.6 Machine data display options

Through the use of the machine data display filter, it is possible to reduce the number of displayed machine data relating to a certain area, e.g. general machine data, for special purposes.

### Machine data areas

Display filters are available for the following machine data areas:

- General machine data
- Channel-specific machine data
- Axis-specific machine data
- Drive machine data
- Display machine data

### Filter options

It is possible to limit the machine data display using the following filtering methods:

- Only display axes that are used
- According to indices
- According to display groups
- Display expert parameters

### Only display axes that are used

If this checkbox is activated, axes that are not used are hidden when scrolling.

### Indices from ... to

The index filter refers to the machine data arrays. On the display, this machine data can be identified by the field index attached to the machine data string.

Example: 10000[index]AXCONF_MACHAX_NAME_TAB

If the index filter is activated, machine data arrays are only displayed in the specified index area.

### Display groups

A display group contains machine data within a machine data area that belongs to the same topic.

**Note**

To find out which display group a machine data item belongs to, refer to the "Display filter" parameter associated with the description of the machine data element in question.

References: List Manual (book 1)

Select/deselect display groups to increase or decrease the number of machine data items displayed for the current machine data area.

### Display expert parameters

If the "Display expert parameters" filter is disabled, only the machine data in a machine data area that is required for the basic functionality of the NC is displayed.

### Procedure




1. Select the "Start-up" operating area.




2. Press the "Mach. data" softkey.




3. Press the softkey of a particular data area, e.g. "General MD".
   The list of general machine data is displayed.



4. Press the ">>" and "Display options" softkeys.
   The window "Display options: ..." for the selected machine data opens.



5. Select the required filter options (according to display groups, indices or expert parameters) by activating the checkboxes.




6. Press the "Select all" softkey to activate all display groups.

- OR -




Press the "Deselect all" softkey.

All checkmarks are removed and you can select individual display groups.




Press the "OK" softkey to save the settings.

# 7.7 Editing machine data and drive parameters

## 7.7.1 Editing hexadecimal values

You can use a bit editor to enter the required values for machine data in hexadecimal format.

### Procedure




1. Select the "Start-up" operating area.




2. Press the "Mach. data" softkey and select the required machine data area (e.g. general machine data).




3. Position the cursor on the entry to be edited.




4. Press the <SELECT> or <INSERT> key.

The "Bit Editor" window opens.

All bits of the machine data item that have a description text are displayed.




5. If you activate the "Display all bits" checkbox, all bits regardless of their data type (8, 16 or 32 bits) are displayed. Activate the relevant bit.




6. Press the "OK" softkey. The setting is checked and an error message may be output.

The window closes and you return to the machine data overview.

The new value is displayed in hexadecimal format.

## 7.7.2 Editing BICO values

You can use the BICO editor to enter BICO values for drive parameters.

**Note**

BICO values can also be entered by copying and inserting or also directly entered the field.

### Procedure




1. Select the "Start-up" operating area.




2. Press the "Mach. data" softkey.




3. Press the "Control unit MD" softkey and select a BICO parameter, e.g. "p738".
4. Position the cursor on the entry to be changed, e.g. BICO 63.2091.0




5. Press the <SELECT> or <INSERT> key.
   The "BICO Editor" window opens.




6. Complete the "Drive object number", "Parameter number", "Bit/Index" and "Parameter value (hex)" fields.

**Note**

The BICO editor also opens if you enter a value other than 0 or 1 for the value to be changed.

## 7.7.3 Editing enum values

For drive parameters, you have the option of editing enum elements (e.g. r2, p3, p4, p97 etc.) using a selection list.

### Procedure




1. Select the "Start-up" operating area.




2. Press the "Mach. data" softkey.




3. Select the desired drive parameter area, e.g. "Control unit MD".

4. Select the desired parameter, e.g. "p3" and position the cursor at the entry to be changed, e.g. "[1] Display, Si...".







5. Press the <SELECT> or <INSERT> key.
   A window containing a list of all defined enum values and their descriptions opens.

6. You can select a value from the list.
   - OR
   You can also directly enter the value as decimal number into the brackets [ ].

## 7.8 Searching for data

You can search for specific machine, setting and display machine data.

#### Search strategies

- When a digit or a digit with an index is entered, an exact search for this parameter is performed.
  Example: If "9" is entered > p9 will be found (but not p99)
- When text is entered, a full-text search is performed: i.e. the term is sought in the relevant table and in the explanations in the footers.

---

**Note**

**Search with place holders**

When searching for machine data, you have the option of using place holders:

- "*": Replaces any character string
- "?": Replaces any character

---

## Searched fields

A search is made in the following fields of the machine data for the character string that has been entered if the "Additionally search in the explanation texts" option is not set:

- Machine data number
- Symbolic name
- Unit

The following fields are additionally searched if the "Additionally search in the explanation texts" option is set:

- Descriptive text
- Bit texts
- Enum texts (only for drives)

## Procedure




1. Select the "Start-up" operating area.




2. Press the "Mach. data" softkey.
3. Select the required area by pressing the relevant softkey.




4. Press the "Search..." softkey and enter the required text or digit in the search screen.
5. If you wish to restrict the search only to the texts in the tables that are displayed, deactivate the "Search explanation texts too" checkbox.




6. Select the search direction required (forwards, backwards) in the "Direction" field.

   The cursor is positioned on the first entry that corresponds to the search term.




7. Press the "Continue search" softkey if this machine, setting or display machine data item is not the one you are looking for.

## Additional search option




Press the "Go to start" softkey to start the search at the first data.




Press the "Go to end" softkey to start the search at the last data.

# 7.9 Managing data

## Overview

The "Manage data" function is used to simplify the commissioning and permits the transfer, backup, loading and comparing of machine, setting, compensation and drive data at the file level. The files are saved in the .tea ASCII format.

The "Manage data" softkey is visible as of access level "Service" (= 2).

The following option fields are available in the "Transfer Machine Data" window:

- Transfer data within the controller
  Only those controller objects are listed that are currently available in the controller.
- Save data to a file
  Additional files with the extensions .acx and .log are created for the SINAMICS parameters:
  - Files with the extension .acx contain the parameter descriptions for the selected drive object.
  - The log file may also contain the warnings and errors that occurred during the conversion from .acx to .tea.
- Load data from a file

  **Note**

  An NCK and drive reset is only triggered automatically for drive objects after loading. A reset must be triggered manually for all other controller objects.

- Compare data
  Several files (.tea) and/or controller objects (axis, channel, servo, infeed, etc.) can be compared at the same time. An online comparison of constantly changing data is not possible because the values are saved temporarily at the time of the comparison.
  Up to eight elements can be taken into the list of the data to be compared.
- Exporting/importing SINAMICS lists
  If hydraulic components are used in the system, for example, you can export or import a valve list (Hydraulic Drive).

A description of the application of valve lists can be found in the following documentation:

## References

SINAMICS S120 System Manual Hydraulic Drive, chapter "Commissioning" > "Importing valve lists"

## Procedure




1. Select the "Start-up" operating area.




2. Press the "Mach. data" softkey.

   The "Manage data" softkey is available in each machine data area, except in "User views".

   - OR -




   Press the "System data" softkey and select the HMI data/Data backups/IB data directory.




3. Press the "Manage data" softkey.

   The "Transfer Machine Data" window opens.




4. Activate the required option filed and press "OK".

### Transfer data within the controller




5. Position the cursor on a controller object and confirm with "OK".




6. In the window that opens, select the target controller object and confirm with "OK".

### Save data to a file




5. Position the cursor on a controller object and confirm with "OK".

6. The "Save Under: Select Archive" window opens and the appropriate storage location is preselected.




   If required, press the "New directory" softkey to generate a new subdirectory. Enter the required name in the "New Directory" window and confirm with "OK".




7. In the "New File" window, enter the required name for the file to be saved and confirm with "OK".

   The stored files can be found in the data tree of the "System data" in the HMI data/Data backups/IB data directory.

### Load data from a file

5. The "Load Trace: Select file:" window opens.




   Position the cursor on a .tea file that is located either in the IB data directory or on a USB flash drive and confirm with "OK".

   Please note: If the source and target drive object numbers differ when loading drive data, you must adapt individual parameters. To do this, press the "Adapt" softkey and edit the parameter values in the window that appears.

### Compare data




5. In the "Compare Data - Controller" window, select an element to be compared from the tree structure of the control objects.

   - OR -

File

Press the "File" softkey to select a .tea file to be compared from the tree structure of the files.

Add to list

6. Press the "Add to list" softkey.

   The list of the data to be compared is displayed in the lower part of the window and the selected element is added.

Delete from list

7. Press the "Delete from list" softkey to remove individual elements from the comparison list.

   - OR -

   Deactivate the checkbox of the selected element in the list of the data to be compared.

Compare

8. When at least two elements are listed, the "Compare" softkey is enabled. Press this softkey to perform the comparison.

# 7.10 User views

## Use

In the "User Views" window you can create and adapt all machine data relevant to a particular functionality. Machine data is individually collated to simplify the user's task.

You can include additional comments when creating and/or editing user views.

**Note**

**Importing user views (compatibility)**

If you have already created user views in HMI-Advanced, copy the *.klb files to /**user**/sinumerik/hmi/template/user_views to use them in SINUMERIK Operate.

## Display in the the "Parameter" operating area

Proceed as follows to show user views also in the "Parameters" operating area:

1. Select the "Start-up" operating area.
2. Press the "Machine data" softkey.
3. Press the "User views" softkey.
4. Press the "Manage view" softkey.
5. Press the "New view" softkey.
6. Activate the checkbox "Display parameter / use setting data", if you also wish to display the new user view in the "Parameter" → "Setting data" → "Data lists".
7. Enter a name for the user view.
8. Confirm with "OK".

## 7.10.1 Creating a user view

### Procedure




1. Select the "Start-up" operating area and press the "Mach. data" softkey.







2. Press the softkeys "User views",
"Manage view" and







"New view".
The "New View" window opens.

3. Enter the desired view name.
Press the "OK" softkey.




If a user view already exists with the same name, you receive a safety prompt as to whether the existing view should be overwritten.




4. Press the "Enter data" softkey.
The "Insert data" selection box opens.




5. Open the selection list with the <INSERT> key, select a machine data area and press the <INPUT> key.








Select a machine data item with the cursor keys or using the "Find" softkey.










6. Press the "Insert before line" or "Insert after line" softkey to insert the selected machine data at the required position in the user view.

**Note:**

You can navigate in the open user view with the mouse and select a line without closing the list box.




7. Press the "Back" softkey to close the list box and return to the user view you want to edit.
   Your entries are automatically stored.




8. Press the "Insert text" softkey to insert any text in the open user view.
   The "Insert Text" input window opens.







9. Enter a text and explanatory description for the text and press the "Insert before line" or "Insert after line" softkey to accept the text.




10. Press the "Back" softkey to save your entries and return to the user view you are editing.

### Selecting a particular machine data item




You can use the Search dialog box to look for a particular data item.

## 7.10.2 Editing the user view

Depending on the row you have selected, you can use the "Properties" softkey to change comments and, in the case of machine data, the data source (channel, axis, drive unit) and view the description and, if required, change.

- Machine data
  Depending on the data type you can select either a fixed or a variable data source for machine data.
  The entry "variable(*)" means that when you can select another data source (channel, axis, drive unit) with vertical softkeys 1- 3 (+, - and direct selection) the value displayed changes to refer to the data source currently selected.
  Alternately, you can select a fixed data source so that the displayed value does not change when you select a different data source.
- Comment texts
  Text and description

### Procedure




1. Select the "Start-up" operating area and press the "Mach. data" softkey.



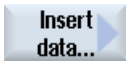




2. Press the "User views" softkey.
   The "User Views" window opens.




3. Press the "Edit view" softkey. Additional softkeys to edit the user view are listed.




4. Press the "Insert data" softkey if you wish to enter another data item.



5. Select a machine data item with the cursor keys.









6. Press the "Up" or "Down" softkeys to move the selected row.

- OR -




Press the "Delete line" softkey to remove the selected line from the view. The data item is removed without a prompt.

- OR -




Press the "Properties" softkey to view the texts for comments and, if required, change.

The "Properties" window is opened. For machine data, depending on the setting, the description or the machine data is displayed in the lower left-hand window.




7. Press the "Back" softkey to save your changes.

## Selecting a particular machine data item




You can use the Search dialog box to look for a particular data item.




Press the "Go to start" softkey to start the search at the first entry.




Press the "Go to end" softkey to start the search at the last entry.

Press the "Continue search" softkey if the data found during the search does not match up with what you are looking for.

### 7.10.3 Deleting a user view

#### Procedure




1. Select the "Start-up" operating area.




2. press the "Machine data" softkey.




3. Press the "User views" softkey.




4. Press the "Manage view" softkey.

5. Select the desired view from the selection list.




6. Press the "Delete view" softkey.




With the final confirmation prompt you will be requested to confirm with either "Yes" or cancel with "No".

or




The "User views" dialog is then redisplayed.

Just like before, as an alternative, it is possible to delete a user view under "System data" under the following path: `../user/sinumerik/hmi/template/user_views`

## 7.11 Plain texts for machine and setting data

Application-specific and language-dependent plain texts can be configured for machine and setting data depending on the version [Index].

### Displaying the plain texts

You set the display of the plain texts using the following display machine data:

| MD9900 $MM_MD_TEXT_SWITCH | Plain texts instead of MD identifier |
|---|---|
| = 1 | Plain text is displayed in the table |
| = 0 | Plain text is displayed in the dialog line |

## Files

| File | Machine/setting data |
|---|---|
| nctea_xxx.txt | General machine/setting data |
| chtea_xxx.txt | Channel-specific machine/setting data |
| axtea_xxx.txt | Axis-specific machine/setting data |
| hmitea_xxx.txt | Display machine data |

For "xxx" the specified language code must be used, e.g. "deu" for German, "eng" for English, etc.
Supported languages (Page 502)

Ensure that the file names are written in lower case letters.

## Procedure

1. Depending on the machine or setting data for which you wish to change the plain text, create the corresponding file, e.g. "nctea_xxx.txt".
2. Save the file with the UTF-8 coding in order that umlauts and special characters are correctly displayed in SINUMERIK Operate.
3. Place the file in SINUMERIK Operate in the following directory:
   System CF card/**user**/sinumerik/hmi/lng or
   System CF card/**oem**/sinumerik/hmi/lng or
   System CF card/**addon**/sinumerik/hmi/lng

   **Note**

   The files are searched through in the sequence of the directories "user", "oem" and "addon". The first file that is found is evaluated. If, e.g. text files are available in the "user" and "oem" directories, the files are evaluated in the "user" directory.

**Note**

If a machine data item has different indices and you do not specify an index for the plain text, the same plain text appears for all the indices of the machine data item.

The parameters and indices must be sorted in an ascending order.

## Example of a file with German texts

```
14510 Mein Integer-Text für PLC_MD 14510

14510[1] Mein Integer-Text für MD 14510[1]

14512[1] Mein Text für HEX-Datum 14512[1]
```

## Result in the display

The plain text is displayed in the table or in the diagnostics line in the operating area "Startup", "General MD".

| Machine data | Text |
|---|---|
| 14510 | My integer text for PLC_MD 14510 |
| 14510[1] | My integer text for MD 14510[1] |
| 14512[1] | My text in HEX data 14512[1] |

# Customizing the "Machine" operating area 8

### Automatic changeover

The circumstances under which an automatic changeover is made into the "Machine" operating area are set in MD51040 $MNS_SWITCH_TO_MACHINE_MASK.

| MD51040 $MNS_SWITCH_TO_MACHINE_MASK | 0 (default setting) |
|---|---|
| Bit 0 = 1 | For program selection in the Program Manager, an automatic change is not made into the Machine operating area. |
| Bit 1 = 1 | For program selection in the Program Manager, an automatic change is not made into the Machine operating area.<br>When changing over the operating mode via the machine control panel, an automatic change is not made into the Machine operating area. |
| Bit 2 = 1 | For program selection in the Program operating area, an automatic change is not made into the Machine operating area. |
| Bit 3 = 1 | When selecting a program/executing a program in the Program operating area, the block search is not automatically started. |

## 8.1 Setting the font size of the actual value display

### Setting font size

The actual values for the machine coordinate system or workpiece coordinate system can be displayed in two different font sizes in the "Machine" operating area.

Using a channel machine data item, you can set how many axes should be displayed in a larger font.

| MD52010 $MCS_DISP_NUM_AXIS_BIG_FONT | Number of actual values with a large font |
|---|---|
| = 3 | Number of actual values with a larger font (default setting) |

If the number of axes changes, e.g. as a result of transformations, then the following channel-specific machine data is relevant for the representation pattern of the font sizes:

| MD52011 $MCS_ADJUST_NUM_AXIS_BIG_FONT | The number of actual values with large font is dynamically adapted to the number of geometry axes |
|---|---|
| = 0 | Only MD52010 is valid. The number is therefore fixed. |
| = 1 | Only the geometry axes are displayed using a large font. MD52010 is ignored. |
| = 2 | The number of geometry axes and content MD52010 are displayed using a large font. |

## 8.2 Inserting a user-specific logo

For the traditional SINUMERIK Operate display, the Siemens logo is visible in the "Machine" operating area in the header.

For the SINUMERIK Operate display of the second generation, the Siemens logo is permanently displayed in the header.

You can replace the Siemens logo with your own logo.

### OEM-specific logo in traditional display

Procedure:

1. Name your own logo "`logo.png`".

   **Note**

   The file name must not contain any lower case letters.

2. Save your own logo depending on the bitmap size in one of the directories: /**oem**/sinumerik/hmi/ico or /**user**/sinumerik/hmi/ico

| Directory | Bitmap size |
|---|---|
| /lco640 | 122 x 19 |
| /lco800 | 160 x 24 |
| /lco1024 | 199 x 30 |

### OEM-specific logo of Generation 2 display

Procedure:

1. Depending on the screen resolution, name your own logo "`company_label_XXXXxYYYY.png`".
   The placeholder XXXXxYYYY corresponds to the screen resolution, for example "`company_label_1024x768.png`".
2. Save your own logo depending on the bitmap size in one of the directories: /**oem**/sinumerik/hmi/skins/skin1/ico/ico640 or /**user**/sinumerik/hmi/skins/skin1/ico/ico640

**Note**

**Display error**

Remove your own logo "`logo.png`" from the directories /oem or /user when the Generation 2 display is used.

Your own logo must not exceed the header height in relation to the screen resolution used.

The following is allowed:

| Screen resolution | Maximum bitmap height |
|---|---|
| 640 x 480 | 32 |
| 760 x 505 | 32 |
| 800 x 480 | 32 |
| 800 x 600 | 40 |
| 1080 x 810 | 53 |
| 1024 x 768 | 51 |
| 1024 x 1014 | 51 |
| 1280 x 768 | 51 |
| 1280 x 800 | 51 |
| 1280 x 1024 | 53 |
| 1300 x 768 | 51 |
| 1366 x 768 | 51 |
| 1440 x 1080 | 72 |
| 1454 x 1080 | 70 |
| 1854 x 1014 | 51 |
| 1854 x 1080 | 70 |
| 1920 x 1080 | 70 |

## 8.3 Configuring the display of the G-code groups

The "G-functions" pane is visible in the "Machine" operating area.

Up to a maximum of 16 G-code groups are displayed in this pane. Configure the "slmagcodeconfig.ini" file accordingly to determine which G-code groups should be displayed, and where.

### Configuring G-code groups

| Section | Meaning |
|---|---|
| Selected G groups | This area is intended for operation of the controller in Siemens mode. |
| Selected G groups ISO mode | This area is intended for operation of the controller in ISO mode. |

Enter the G-code group and position as follows:

| SelectedGGroupN = M | |
|---|---|
| N | Position where G-code group M will be displayed.<br>If you want to leave a position empty, omit the number (M), e.g. "`SelectedGGroup1 = `" |
| M | The number of the G-code group to be displayed at position N |

## Structure of the file

```
[Selected G-Groups]
SelectedGGroup1 = 1
SelectedGGroup2 = 2
SelectedGGroup3 = 3
SelectedGGroup4 = 4
SelectedGGroup5 = 5
SelectedGGroup6 = 6
SelectedGGroup7 = 7
SelectedGGroup8 = 8
SelectedGGroup9 = 9
SelectedGGroup10 = 11
SelectedGGroup12 = 12
SelectedGGroup13 = 13
SelectedGGroup14 = 14
SelectedGGroup15 = 15
SelectedGGroup16 = 16
[Selected G-Groups ISO-Mode]
... (as with section for Siemens)
```

## Procedure

1. Copy the "slmagcodeconfig.ini" file from the folder /**siemens**/sinumerik/hmi/template/cfg.
2. Place the file in the following directory: /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg. As soon as the file is located in the user-specific directory, the entries for this file take precedence over the Siemens file. If an entry is missing in a user-specific file, the corresponding entry from the Siemens file is used instead.
3. Open the file and enter the number of the G code in question at the required position "`SelectedGGroup1 =` **5**".
   I.e. the 5th G code group will be displayed at the 1st position.

# 8.4 Configuring the channel operational message

## Overview

The channel operational messages are displayed in the third line of the header in the "Machine" operating area. There are two types of channel operational messages:

1. Channel operational messages that require an operator action to continue a program, e.g. feed enable missing, M01 active.
   
   The channel operational messages are identified by exclamation marks.
2. Channel operational messages that do not require an operator action to continue a program. A stop condition is active for a length of time (>1 s) and is interrupted by the program processing, e.g. dwell time active, waiting for tool change.
   
   The channel operational message is identified by a clock icon and disappears after a certain time.

## File

The settings for this are specified in the "slmahdconfig.ini" file.

## Configuring the channel operational message

| Section | Meaning |
|---|---|
| CondStopIcons | Specification of an icon that is to be displayed with a certain channel operational message.<br>If no icon is specified for a certain channel operational message, then the icon specified for "DefaultIcon" is displayed. |
| CondDelayTime | Specification of how long (in milliseconds) a certain message must be present before it is displayed.<br>• If no time is specified for a certain channel operational message, then the time specified at "DefaultDelayTime" is used.<br>• If the channel operational message is to be displayed immediately, "0" must be specified as time. |

## Procedure

1. Save your own icon files under the following path in one of the intended subdirectories:
   /**oem**/sinumerik/hmi/ico or
   /**user**/sinumerik/hmi/ico
   Depending on the screen resolution, there are the following subdirectories:

   | Directory | PNG image size (pixels) |
   |---|---|
   | /Ico640 | 16 x 16 |
   | /Ico800 | 17 x 17 |
   | /Ico1024 | 27 x 27 |

2. Copy the sample file "slmahdconfig.ini" from the directory
   /**siemens**/sinumerik/hmi/template/cfg.

3. Place the file in the following directory:
   /**oem**/sinumerik/hmi/cfg or
   /**user**/sinumerik/hmi/cfg
4. Open the file and make the required settings:
   - Specify an individual icon by changing the file name of the icon for channel operational messages.
   - Specify a different period by changing the time value for the channel operational message.
5. Save the settings.
   The configurations are activated after booting.

## Example of file "slmahdconfig.ini"

### Configuration

The following default messages are intended when technical faults occur:

```
[CondStopIcons]
DefaultIcon = condwait.png
1 =  condstop.png    ; No NC ready
2 =  condstop.png    ; No mode group ready
3 =  condstop.png    ; Emergency stop active
4 =  condstop.png    ; Alarm with stop active
...
26 = condwait.png    ; Waiting for position control
...
30 = condwait.png    ; Waiting for safe operation
31 = condwait.png    ; No channel ready
...
46 = condwait.png    ; Rapid retraction started
...


[CondDelayTime]
DefaultDelayTime=1000
1 = 0           ; No NC ready
2 = 0           ; No mode group ready
3 = 0           ; Emergency stop active
4 = 0           ; Alarm with stop active
...
```

## 8.5 Deactivating program test

There are several program test options to test and run-in a part program that are executed using the "Program Test" function.

This function is active as standard. You can select the program test with the "PRT" check box in the "Program control" window in the "Machine" operating area.

### Deactivating the program test

You can deactivate the program test using the following configuration machine data. "PRT" is no longer displayed in the status line.

| MD51039 $MNS_PROGRAM_CONTROL_MODE_MASK | | Options for machine - program control |
|---|---|---|
| Bit 0 = 0 | Function program test not available | |

The test option is described in the following documentation:

### References

Function Manual, Basic Functions, Mode Group, Channel, Program Operation, Reset Behavior (K1)

## 8.6 Display STAT and TU

### Display STAT and TU in the "WCS" window

If you want to display the joint position of the machine with the value STAT and the position of the rotary axes with the value TU in the "Machine" operating area for Cartesian point-to-point or PTP traversing, set the following channel-specific configuration machine data ≠ 0:

| MD52032 $MCS_STAT_DISPLAY_BASE | Number base for display of the joint position STAT |
|---|---|
| = 0 | No display, value from MD51032 effective (default) |
| = 2 | Display as binary value |
| = 10 | Display as decimal value |
| = 16 | Display as hexadecimal value |

| MD52033 $MCS_TU_DISPLAY_BASE | Number base, display of the rotary axes TU position |
|---|---|
| = 0 | No display, value from MD51033 effective (default) |
| = 2 | Display as binary value |
| = 10 | Display as decimal value |
| = 16 | Display as hexadecimal value |

The settings in MD52032 and MD52033 also work for Teach In.

For further information about importing STAT and TU under Teach In, see Chapter "Activating the "Teach In" function (Page 126)".

A description of the function and programming is provided in the following documentation:

### References

Extended Functions Function Manual; Kinematic Transformation (M1)
Section "Cartesian PTP travel."

## 8.7 Activating the "Teach In" function

### Activate Teach In

The Teach In function is activated in part via the general configuration machine data.

### Settings

| MD51034 $MNS_TEACH_MODE | |
|---|---|
| Bit 0 = 1 | The "Progr. teach in" softkey is displayed. The approached position is taken into the program with the "Accept" softkey. (default) |
| Bit 1 = 1 | The acceptance of the teach-in block can be blocked by the PLC.<br>DB19.DBX13.0 = 0 Block is accepted<br>DB19.DBX13.0 = 1 Block is not accepted |

Access to the function still depends on access rights.

| MD51045 $MNS_ACCESS_TEACH_IN | Protection level TEACH IN |
|---|---|
| = 7 | The protection level for the execution of traversing movements to MDA buffer / part program for TEACH IN is specified. (default) |

| MD51050 $MNS_ACCESS_WRITE_PROGRAM | Protection level write part program |
|---|---|
| = 7 | The protection level for the writing of part programs is specified. (default) |

Set the channel-specific machine data:

| MD20700 $MC_REFP_NC_START_LOCK | NC-Start disable without reference point |
|---|---|
| = 0 | |

| MD52207 $MCS_AXIS_USAGE_ATTRIB | Attributes of the axes<br>[n] channel axis number |
|---|---|
| Bit 14 | Activate spindle in axis mode for Teach In |
| = 1 | A spindle in axis mode should also be teachable |

## Activating Teach In

1. Set the machine data and initiate a reset.
2. Now switch to the AUTO and MDA operating mode.
3. Select the TEACH IN machine function.
4. Press the "Teach prog." softkey.

## Teach In not active

If the "Prog. teach in" softkey is not active, the reasons can be as follows:

- The program is not in the stop, reset or interrupt state.
- Not all of the axes that require referencing are referenced.

## Accept STAT and TU

If for the Cartesian point-to-point or PTP travel you want to import the joint position with the value STAT of the machine and the position of the rotary axes with the value TU into the traversing block for Teach In, click on the checkboxes STAT and TU in the "Settings" window under "Prog. teach in".

The checkboxes are only displayed if the following general configuration machine data is set to ≠ 0:

| MD52032 $MNS_STAT_DISPLAY_BASE | Number base for display of the joint position STAT |
|---|---|
| = 0 | no display |
| = 2 | Display as binary value (default) |
| = 10 | Display as decimal value |
| = 16 | Display as hexadecimal value |

| MD52033 $MNS_TU_DISPLAY_BASE | Number base, display of the rotary axes TU position |
|---|---|
| = 0 | no display |
| = 2 | Display as binary value (default) |
| = 10 | Display as decimal value |
| = 16 | Display as hexadecimal value |

# 8.8 Block search

## 8.8.1 Activating block search mode

### Block search mode settings

The following machine data can be used to set the possible variants for the block search with G-code programming:

| MD51028 $MNS_BLOCK_SEARCH_MODE_MASK | Bit mask for the available search modes |
|---|---|
| Bit 0 = 1 | Block search with calculation, without approach |
| Bit 1 = 1 | Block search with calculation, with approach |
| Bit 4 = 1 | Block search without calculation |
| Bit 5 = 1 | Block search with test run |

The following machine data can be used to set the possible variants for the block search in a JobShop program (ShopMill, ShopTurn):

| MD51024 $MNS_BLOCK_SEARCH_MODE_MASK_JS | Bit mask for the available search modes (ShopMill, ShopTurn, single-channel) |
|---|---|
| Bit 0 = 1 | Block search with calculation, without approach (default setting) |
| Bit 1 = 1 | Block search with calculation, with approach<br>Notice:<br>In a ShopMill/ShopTurn program, the search is performed only on G code-blocks. |
| Bit 3 = 1 | Skip EXTCALL programs |
| Bit 5 = 1 | Block search with test run |

| MD52212 $MCS_FUNCTION_MASK_TECH | General function screen for all technologies |
|---|---|
| Bit 3 | Block search cycle for ShopMill / ShopTurn |
| = 0 | The E__ASUB or F_S_ASUB cycles are not called in the block search cycle PROG_EVENT.SPF |
| = 1 | The E__ASUB (under ShopMill) or F_S_ASUB (under ShopTurn) cycles are called in the block search cycle PROG_EVENT.SPF |
| Bit 5 | Call block search function for SERUPRO |
| = 0 | CYCLE207 is not called in the block search cycle PROG_EVENT.SPF. |
| = 1 | In the block search cycle PROG_EVENT.SPF (for block search cycle with program test - SERUPRO), the cycle for SERUPRO (CYCLE207) is called. After block search with program test (SERUPRO), cycle CYCLE207 triggers the output of the collected machine commands. |
| Bit 15 | Do not set $MC_TOOL_CARRIER_RESET_VALUE after block search |
| = 0 | In the block search cycle Prog_Event.SPF, the current tool holder is written into the machine data $MC_TOOL_CARRIER_RESET_VALUE. |
| = 1 | In the block search cycle Prog_Event.SPF, the current tool holder is not written into the machine data $MC_TOOL_CARRIER_RESET_VALUE. |

| | |
|---|---|
| Bit 16 | Do not automatically execute swivel plane after block search |
| = 0 | In the block search cycle PROG_EVENT.SPF, the section for standard cycles: Swivel plane is executed |
| = 1 | In the block search cycle PROG_EVENT.SPF, the section for standard cycles: Swivel plane is not executed |
| Bit 17 | Do not automatically align turning tool after block search |
| = 0 | In the block search cycle PROG_EVENT.SPF, the section for aligning the turning tool is executed. |
| = 1 | In the block search cycle PROG_EVENT.SPF, the section for aligning the turning tool is not executed. |

**Note**

The "Block search mode" softkey is only displayed, if, in the machine data MD51024 or MD51028, more than one version for the block search has been defined; for example "Block search with calculation without approach" and "Block search with calculation with approach".

When so doing, take into account that the search versions set active in the "Block search mode" window are also referred to the program.

### 8.8.2 Accelerated block search for execution from external

The accelerated block search function allows you to run the part program forward until you reach the block you require.

#### Settings

If you activate the "Skip EXCALL" function, then you must set the following general configuration machine data:

| | |
|---|---|
| MD51028 $MNS_BLOCK_SEARCH_MODE_MASK | Bit mask for the available search modes |
| Bit 3 = 1 (33H to 3BH) | |

For an external main program, a block search without calculation is always a fast block search. Only the blocks from the search position are loaded.

## 8.9 Multi-channel support

In the "Machine" operating area, up to any four channels of a mode group can be displayed side by side. The number of channels is dependent on the display size and resolution of the operator panel used.

Using the multi-channel editor, for a multi-channel machine, you can open, change or generate programs simultaneously that are independent of one another from a time perspective. In this case, the programs involved are displayed next to one another in the editor.

Further, you have the option of simulating the programs before they are actually processed at your control.

In addition, you can align the programs according to synchronization points.

## View options of the operator panel

The operator panel supports the following views:

| Operator panel | Maximum number of channels |
|---|---|
| OP12 | 2 channels |
| OP15 | 3 channels |
| OP15 black | 4 channels |
| OP19 | 4 channels |
| TOP 1500 | 4 channels |
| TOP 1900 | 4 channels |
| TOP 2200 | 4 channels |

## Precondition

- Multi-channel data only involve machines where the first technology is turning.
- For multi-channel machine, the following data must be the same for all channels:
  - Measurement unit
  - Zero offset (e.g. G54)
  - Z value of the zero offset (optional)
  - Blank
  - Speed limitation
- Further, the following JobShop-specific data must be the same for all channels:
  - Retraction plane (one each for the main and counterspindle)
  - Tailstock (one each for the main and counterspindle)
  - Tool change point (one each for the main and counterspindle)
  - Safety clearance
  - Climbing and conventional

## Creating a multi-channel program

For multi-channel machining, you require a job list with the following data:

- Program names
- Program type
- G code

- ShopTurn program
- Channel assignment



**Software option**

You require the "ShopMill/ShopTurn" option to generate and edit ShopTurn/ ShopMill programs.

After you have created a job list, in the "multi-channel data" parameter screen, you can combine the data from the G code and JobShop, which is then saved as cycle call in the job list. The "multi-channel data" parameter screen is displayed as soon as at least one ShopTurn program is contained in a job list.

### Only programming G code programs

If the job list only contains G code programs, then you can deactivate the "multi-channel data" function. To do this, set the following machine data:

| MD51228 $MNS_FUNCTION_MASK_TECH | General function screen for all technologies |
|---|---|
| Bit 0 | G code programming without multi-channel data |
| = 1 | Multi-channel data is not listed for job lists that only contain G code programs. The "Multi-channel data" softkey is not displayed. |

## 8.10 Manual machine

### General information

"Manual machine" allows you to work with large cycles in manual mode without having to write programs especially for this purpose.



**Software option**

You require the following software option in order to use this function:

"ShopMill/ShopTurn"

### Activation

| MD52212 $MCS_FUNCTION_MASK_TECH | Cross-technology function mask |
|---|---|
| Bit 8 | Manual machine (ShopMill/ShopTurn) |

### Enables

| MD52214 $MCS_FUNCTION_MASK_MILL | Milling function screen |
|---|---|
| Bit 5 | Enable spindle control, tool spindle |
| = 0 | via the NC/PLC interface |
| = 1 | via the user interface |

| | |
|---|---|
| Bit 6 | Enable spindle control, rotating spindle |
| = 0 | via the NC/PLC interface |
| = 1 | via the user interface |

| MD52218 $MCS_FUNCTION_MASK_TURN | Turning function screen |
|---|---|
| Bit 3 | Enable spindle control, main spindle |
| = 0 | via the NC/PLC interface |
| = 1 | via the user interface |
| Bit 4 | Enable spindle control, counterspindle |
| = 0 | via the NC/PLC interface |
| = 1 | via the user interface |
| Bit 5 | Enable spindle control, tool spindle |
| = 0 | via the NC/PLC interface |
| = 1 | via the user interface |

| | |
|---|---|
| SD43200 $SA_SPIND_S | Speed for spindle start using NC/PLC interface signal; value outputs using the operating software |
| SD43202 $SA_SPIND_CONSTCUT_S | Cutting speed for spindle start using NC/PLC interface signal; value outputs using the operating software |
| SD43206 $SA_SPIND_SPEED_TYPE | Spindle speed pipe for spindle start using NC/PLC interface signal; value outputs using the operating software |

## Additional parameter assignments

We recommend the following additional parameter assignments:

| Handwheel | |
|---|---|
| MD11346 $MN_HANDWH_TRUE_DISTANCE = 3 | Handwheel path or velocity specification |
| MD11602 $MN_ASUP_START_MASK bit 0, 3 =1 | Ignore stop conditions for ASUB |
| MD20150 $MC_G_CODE_RESET_VALUE [7] = 2 | Delete position of the G groups; run-up G54 active |
| MD20624 $MC_HANDWH_CHAN_STOP_COND = 0xFFFF | Definition of the behavior of traveling with handwheel, channel-specific |
| MD32084 HANDWH_STOP_COND = 0x7FF | Behavior, handwheel travel |
| MD52212 FUNCTION_MASK_TECH Bit 9 = 1 | Function screen form across technologies; work offset via softkey |

**Note**

If, in machine data MD52212 FUNCTION_MASK_TECH bit 9 is set, the "WO selection" softkey is displayed in the "Work Offset – G54 … G599" window.

| Working area limits | |
|---|---|
| MD28600 $MC_MM_NUM_WORKAREA_CS_GROUPS > 0 | Number of coordinate system-specific working area limits |
| MD20150 $MC_GCODE_RESET_VALUES[59] = 2 | Delete position of the G groups; activate working area limits of the group |

**Note**

If $MC_GCODE_RESET_VALUES[59] = 2 is set in the MD20150 machine data, the "Limit stops" softkey is displayed in the "Manual Machine" window. This requires that $MC_MM_NUM_WORKAREA_CS_GROUPS also has the value 1 for MD28600.

| Feedrate | |
|---|---|
| SD42600 $SC_JOG_FEED_PER_REV_SOURCE = -3 | Revolutional feedrate control in JOG mode |
| SD43300 $SA_ASSIGN_FEED_PER_REV_SOURCE [X axis] = -3 | Revolutional feedrate for positioning axis X |
| SD43300 $SA_ASSIGN_FEED_PER_REV_SOURCE [Z axis] = -3 | Revolutional feedrate for positioning axis Z |

**Note**

Ensure that the axis-specific setting data SD43300 SA_ASSIGN_FEED_PER_REV_SOURCE are set the same.

### References

- Operating:
  Operating Manual, Turning/Milling, Chapter "Working with a manual machine"
- Programming:
  Programming Manual, Fundamentals; Chapter "Working area limit"

## 8.11 User status display (OEM)

### Function

User icons can be displayed in the Machine operating area in the second line in the header. In this case, the program name is displayed in the right-hand field of the third line in the header. Active hold conditions hide the program name.

The display of the user icons is controlled via PLC bits. The icons are provided in the form of PNG files and stored resolution-specifically in the file system at: `/user/sinumerik/hmi/ico/ico640`, etc. **Maximum 16 positions are available in the header** to display the user icons.

## Configuration

The user icons are configured in the `slmahdconfig.ini` file in Section [UserIcons]. The following data is required:

- The start address of the PLC double word with the bits to control the display of the user icons.
- For each bit used, the file name of the icon and the position at which the icon should be displayed.

Alternatively, you can use the following parameters for the definition:

- **USER_ICON_BASE**
  This means that 32 bits are available with UI_0 (highest bit) to UI_31 (lowest bit).
  Example: USER_ICON_BASE=DB80.DBB0
  ⇒ UI_0 = DB80DBX3.0 and UI_31 = DB80.DBX0.7
- **USER_ICON_BASE256**
  This means that 256 bits are available with UI_0 (highest bit) to UI_255 (lowest bit).
  Example: USER_ICON_BASE256=DB19.DBB80
  ⇒ UI_0 = DB19.DBX111.0 and UI_255 = DB19.DBX80.7

## Example

```
[UserIcons]
USER_ICON_BASE=DB19.DBB80
; set the start byte of the PLC double word that displays the icons (DB19.DBD80)
UI_0 = icon1.png,4
; show icon1.png on position 4 if bit 0 (DB19.DBX83.0) is set
UI_1 = icon2.png,6
; show icon2.png on position 6 if bit 1 (DB19.DBX83.1) is set
```

## Superimposition

With a **non-unique** selection (bits for multiple identifiers whose image is to be displayed at the same position):

The image with the lower identifier number is superimposed with the image with the higher identifier number for the same position. Non-unique selection can also be used with less than 16 active selection bits. The PLC can deliberately use this superimposition to overlay less important displays (with lower identifier numbers) with important displays (high identifier numbers).

**Note**

If two HMI applications are assigned to one NC/PLC unit (M:N), the PLC must first determine from the interface which of the HMI interfaces is active. Icons can be requested by the PLC for the active HMI application only.

## Possible applications

With a **unique selection**, a maximum of 16 bits are set simultaneously in the four selection bytes for each of the different positions.

- 32 alternative images for a position, each selected with a different bit (currently only one active selection bit)
- Two alternative images for each of the permissible 16 positions, each position requires two of the 32 selection bits (maximum 16 active selection bits at any one time)
- Combination thereof with a maximum total of 32 images for the maximum 16 positions (maximum of 16 active selection bits at any one time)
- More than 16 selection bits.

## Configuring OEM texts

Machine manufacturers can display their own texts in the program directory and program name lines. These texts cannot include % parameters as is possible in user texts. The lines are divided into <OEM_NUMBER_TEXTFIELD> areas according to the entry in HEADER.INI.

The data is provided in a user data block. The number of the data block and the exact offset are declared in the HEADER.INI file. The following entries are available in the [OEMTextInfo] section for this purpose:

```
[OEMTextInfo]
OEM_NUMBER_TEXTFIELD = 1 ; maximum of 3 text fields per line
OEM_TEXT_BASE = DBx.DBBy
```

The interface in the user DB has the following format:

| | |
|---|---|
| Start-DBB byte<br>or<br>Start-DBB +18 byte | Bit0=1: there is something to do<br>All bits=1: job is acknowledged by HMI<br>All bits=0: PLC can write to interface again. |
| Start-DBB +1 word<br>or<br>Start-DBB +19 word | Offset for text from alpu_xx.com to be displayed;<br>700 000 is the base address and is always summed internally.<br>All bits=0: Delete line |
| Start-DBB + 3 string<br>or<br>Start DBB+20 double Word | String containing a maximum of 12 characters |

The first parameter contains an offset for a text number with a base of 700,000 (user PLC texts). If this parameter = 0, the corresponding line is deleted. The second parameter contains a maximum 12-character string. Unless empty, this text is output instead of the user PLC text.

The parameters must be written from "back to front" in the PLC, i.e. all parameters must be initialized before the first byte is set. The data for the next line is located exactly 18 bytes after the start byte. The lines are output from left to right; a maximum of three output fields per line are provided (i.e. a total of six fields are possible).

A hotlink to the first byte for each line is set up in the operating software. If a change occurs in that byte, the operating software responds. If the operating software displayed the data, it writes a -1 to the first variable. The operating software is informed about this change by means of the hotlink, whereupon it writes a 0 to this variable.

The HMI itself does not have to react to this change. The PLC can only write data to the interface again if the variable is set to 0. This mechanism ensures that the operating software knows about all parameter changes.

### Configuring OEM icons

OEM icons indicating the machine state can be output in the field provided for the program control display (e.g. SBL, M01, etc.). If OEM icons are defined, the elements for the program control display are hidden.

The icon names are identified in the HEADER.INI configuration file, similar to user icons:

```
[OemIcons]
 OI_0= <name.png>, <position>
 ...
 OI_31= <name.png>, <position>
```

Here, **<name>** is the file name of the bit map and **<Position>** is the output position (from 1 to 16) in the display line. Multiple bit maps can be output at the same position. If more than one bit map is active simultaneously at the same position, the bit map with the highest screen number is displayed.

The output is controlled by means of a PLC double word. This double word is declared in the next section in the HEADER.INI file:

```
[OemIcons]
 Oem_ICON_BASE = DBx.DBBy
```

Each bit in this double word represents exactly one OEM icon corresponding to the screen number. Thus, if bit 0 is set, bit map OI_0 is displayed. If a bit is reset, the associated bit map is deleted.

A maximum of 16 icons can be displayed, enabling a total of 16 display positions. Empty positions need not be specified.

## 8.12 Configuring the offset display

### Configuring the screen display

Overrides resulting from handwheel offsets (DRF) or tool length corrections from the value of system variable $AA_TOFF can be directly displayed on the screen.

The screen display for the override is configured using the following channel-specific machine data:

| MD52210 $MCS_FUNCTION_MASK_DISP | Function screen display |
|---|---|
| Bit 14 | Display DRF offset under Machine |
| Bit 15 | Display tool offset ($AA_TOFF) under Machine |

## 8.13 Activate machining time recording

If you want to record and store the machining time in Automatic mode, you can make the settings on the user interface.

You will find additional information about making these settings in the following document:

### References

Universal/Turning/Milling/Grinding Operating Manual

- Section "Machining a workpiece" > "Settings for Automatic mode mode"
- Section "Machining a workpiece" > "Editing a program" > "Settings for the editor"

### Switching on the time recording

To use the function, the following channel machine data must be so set in all channels, even if the machining time recording is not activated for all channels:

| MD18370 $MN_MM_PROTOC_NUM_FILES[1] | Maximum number of log files |
|---|---|
| = 2 | For OEM applications |

| MD18371 $MN_MM_PROTOC_NUM_ETPD_STD_LIST[1] | Number of standard ETPD data lists |
|---|---|
| = 6 | Number of standard data lists in the OPI module ETPD that are needed for Trace system functions. |

| MD28300 $MC_MM_PROTOC_USER_ACTIVE[1] | Activate logging for one user |
|---|---|
| = 1 | For system functions |

| MD28302 $MC_MM_PROTOC_NUM_ETP_STD_TYP[1] | Number of ETP standard event types |
|---|---|
| = 5 | Number of standard event types in the OPI module ETP that are needed |

**Note**

Machine data MD18370, MD18371, and MD28302 are still valid for machining time recording when they are assigned higher values.

# Simulation and simultaneous recording 9

## 9.1 Simulation overview



**Software option**

You require the "3D simulation 1 (finished part)" option for the 3D representation of the simulation.

Simulation is supported with its own program interpreter (SNCK) and a separate simulation-data environment in SINUMERIK Operate. The SNCK considers the complete syntax of the SINUMERIK control family, including the possibility of incorporating special user options on the machine by comparing data with the NC environment. The simulation data can be matched statically as required with the NC environment (initialization data, macros, user data, tool data, machining cycles) or also dynamically when tool data or machining cycles are changed.

Machining simulations, with emphasis on the drilling, milling, turning and grinding technologies, can be performed in the workpiece coordinate system for certain machine kinematics on the user interface of the controller:

- The simulation of the finished part is performed with the real NC data.
- The NC data is automatically compared at each change of the part program.
- The simulation allows a dynamic representation of the machining, even for active 5-axis transformation (TRAORI) and swiveled planes.
- Turning on milling machines.
- Simulation of a machining channel for conventional milling machines.
- Simulation of up to four machining channels for turning machines with B axis.
- Intelligent determination of the block times and the program execution time.
- Very fast graphical representation through the continuous refinement of the workpiece.
- Optimum resolution for each selected picture area.
- Any sections are possible.
- In parallel to the machining of one workpiece, the machining of another workpiece can be simulated (as of NCU 720.x).
- 3D representation of simulation (option).

### General conditions for the simulation

- Very fast simulation result, this means, however, low dynamic resolution.
- For complex representations, fine recording is too slow or does not run for very complex parts.
- "Quickview" for mold-making part programs.
- The simulation cannot be used for compile cycles with special kinematics.

- Compile cycles are supported only to a limited extent.
- Blank CAD data cannot be imported (for example, for castings).
- Couplings cannot be simulated (coupled motion, axial master-value coupling, electronic gearbox):
  - Coupled motion: TRAILON(<following axis>, <leading axis>, <coupling factor>)
  - Axial master-value coupling: LEADON(Faxis, LAxis, curve table n) actual value and setpoint coupling
  - Electronic gearbox: EGDEF(FA, LA1, coupl.1, LA2, coupl.2, ... LA5, coupl.5), EGON, EGONSYN, EGONSYNE
- Reference point approach G74 is not operative during simulation
- The following is not supported:
  - Axis container
  - Dependencies on part programs and PLC
  - Multiple clamping
  - CNC function Synchronized actions for safe programmable logic (SPL) in the NC
- Different technologies in different channels with different axis coordinate systems are not fully supported.
- A 3D simulation is not possible for the grinding technology.
- The software limit switches are not evaluated:
  In a manufacturer cycle for the tool replacement, the "Software limit switch + the Z axis" machine data is used as replacement position:
  G0 Z=$MA_POS_LIMIT_PLUS[Z]-0.1
  However, not this position but the machine data default setting = 1e8 = 100,000,000 is used in simulation. This default position is traversed in the simulation. This makes the simulation very slow and causes the machining time to explode, e.g. 555 hours rather than 2 minutes.
- Program start in the simulation:
  If MD22622 $MCS_DISABLE_PLC_START is set in a channel, then the channel is not automatically started in the simulation. It can then only be started from another channel using the part program command "START". This function is not available for the SINUMERIK 828D.
- Alarm display in the simulation:
  Alarms are output in the simulation exactly the same way as alarms in the normal program execution, but no more than five alarms are displayed (above the graphic, to distinguish them from the machine alarms).
  If a different error signaling behavior (e.g. syntax error) occurs between simultaneous recording and simulation, then, for example, the reason can be due to the hardware access operations in the programming. The simulation does not access the hardware, however, internally it reads the input with 0 and is then exited. On the other hand, for simultaneous recording, an alarm is output.

# 9.2 Setting the technology for simulation

## Turning technology

Examples of machine kinematics:

- Conventional turning with two geometry axes
- Three spindles: Main spindle, counterspindle, tool spindle
- Counterspindle slides, tailstock as NC axis
- B axis: Aligning turning tools in the tool spindle
- Milling with geometry axes: TRANSMIT, TRACYL, TRAANG

Machine data for the turning technology:

| | |
|---|---|
| MD52200 $MCS_TECHNOLOGY = 1 | Turning technology |
| MD52000 $MCS_DISP_COORDINATE_SYSTEM = 34 | Position of the coordinate system (example) |

## Milling technology

Examples of machine kinematics:

- Milling with five axes: Swivel/TRAORI
- Swivel head change

Machine data for the milling technology:

| | |
|---|---|
| MD52200 $MCS_TECHNOLOGY = 2 | Milling technology |
| MD52000 $MCS_DISP_COORDINATE_SYSTEM = 0 | Position of the coordinate system (example) |

## Grinding technology

Machine data for the grinding technology:

| | |
|---|---|
| MD52200 $MCS_TECHNOLOGY = 3 | Cylindrical grinding technology |
| MD52200 $MCS_TECHNOLOGY = 4 | Surface grinding technology |
| MD52000 $MCS_DISP_COORDINATE_SYSTEM = 0 | Cylindrical grinding and surface grinding with moving column (example) |
| MD52000 $MCS_DISP_COORDINATE_SYSTEM = 6 | Surface grinding with table machine (example) |

## Meaning of the axes

| MD52206 $MCS_AXIS_USAGE[n] | Meaning of the axes in the channel<br>[n] channel axis number |
|---|---|
| = 0 | No special meaning |
| = 1 | Tool spindle (driven tool) |

| | |
|---|---|
| = 2 | Auxiliary spindle (driven tool) |
| = 3 | Main spindle (turning) |
| = 4 | C axis of the main spindle (turning) |
| = 5 | Counterspindle (turning) |
| = 6 | C axis of the counterspindle (turning) |
| = 7 | Linear axis of the counterspindle (turning) |
| = 8 | Tailstock (turning) |
| = 9 | Back rest (turning) |

**Note**

To prevent errors occurring, the same axis must be defined as main or counterspindle for all channels.

Enter the direction of rotation for the rotary axes that are not configured in a tool holder or a 5-axis transformation via the following channel-specific machine data.

| MD52207 $MCS_AXIS_USAGE_ATTRIB[n] | Attributes of the axes<br>[n] channel axis number |
|---|---|
| Bit 0 | Rotates around 1st Geometry axis (for rotation axes) |
| Bit 1 | Rotates around 2nd Geometry axis (for rotation axes) |
| Bit 2 | Rotates around 3rd Geometry axis (for rotation axes) |
| Bit 3 | Reversal of direction of rotation (for rotary axes) |

The MD52290 $MCS_SIM_DISPLAY_CONFIG machine data acts only on the OP019. It has no significance for SINUMERIK 828D.

| MD52290 $MCS_SIM_DISPLAY_CONFIG | Position of the status display of the channel in the simulation. |
|---|---|
| Bit 0 | Upper left-hand corner |
| Bit 1 | Upper right-hand corner |
| Bit 2 | Lower left-hand corner |
| Bit 3 | Lower right-hand corner |

In the channel-specific machine data, for at least one geometry axis per channel, enter a value other than 0.0, e.g. 0.001. If the value is 0.0, the system assumes that this parameter has still not been set.

| MD53230 $MCS_SIM_START_POSITION | Axis position when starting the simulation |
|---|---|
| Simulation is only possible if a value not equal to 0 has been set for at least one geometry axis. | |

## Deactivating simulation

The following machine data must be set to deactivate the simulation:

| MD51226 $MNS_FUNCTION_MASK_SIM | Simulation function screen |
|---|---|
| Bit 1 = 1 | Deactivate simulation |
| Bit 10 = 1 | Hide zero point symbol |

### Tool change time

The following machine data defines the estimated time for a tool change in the simulation:

| MD10190 $MN_TOOL_CHANGE_TIME | Tool change times for simulation |
|---|---|
| = 0 | Seconds (default value 0) |

The value only replaces times that are principally not incurred for a simulated tool change, e.g. wait times for PLC commands (M206).

Traversing commands are added to the calculated time for the duration of the simulation if the commands are included in the tool change cycle.

Add the following example to the programming in the tool change cycle, which means that the value from MD10190 can be incorporated as total tool change time in the estimated simulation time:

```
IF $MN_SIM_ENVIRONMENT B_AND 'B100'
 M206 ; see MD 22560 $MC_TOOL_CHANGE_M_CODE
 RET
ENDIF
```

## 9.3 Simultaneous recording overview



**Software option**

You require the option "Simultaneous recording (real-time simulation)" for the "Simultaneous recording" function.



**Software option**

You require the "ShopMill/ShopTurn" option to display the tailstock



**Software option**

You require the "3D simulation 1 (finished part)" option for the 3D representation of the simultaneous recording function.

During machining, the tool paths can be simultaneously recorded on the display of the controller in a three-side view or 3D view. Workpiece graphics and views correspond to the graphic simulation.

**Note**

To prevent an incomplete representation, you must activate the simultaneous recording prior to the NC start.

### Example: Activating the tailstock representation in the spindle chuck data

Procedure:

1. Make the tailstock visible by making the selection in the program.
2. Specify the tailstock dimensions:
   Select in the operating area "Parameters" → "Setting data" → "Spindle chuck data", and enter the "ZR" and "XR" parameters for the tailstock length and tailstock diameter.
   The tailstock dimensions are written into the following machine data:

| MD52218 $MCS_FUNCTION_MASK_TURN | Turning function screen |
|---|---|
| Bit 2 = 1 | Tailstock enable:<br>The tailstock becomes visible if selected in the "tailstock" program. |

| MD53240 $MAS_ SPINDLE_PARAMETER[i] | Spindle chuck data in mm |
|---|---|
| [0]: Chuck dimension<br>[1]: Stop dimension<br>[2]: Jaw dimensions | |

| MD53241 $MAS_ SPINDLE_CHUCK_TYPE | Spindle jaw type |
|---|---|
| 0 = clamp from the outside<br>1 = clamp from the inside | |

| MD53242 $MAS_TAILSTOCK_PARAMETER[i] | Tailstock data in mm |
|---|---|
| [0]: Tailstock dimension<br>[1]: Tailstock length | |

### Activating the machining time display

Timers are provided as system variables in the Program runtime function. While the NC-specific timers are always activated (for time measurements since the last controller power-up), the channel-specific timers must be started via the following channel-specific machine data.

| MD27860 $MC_PROCESSTIMER_MODE | Activate the program runtime measurement. |
|---|---|
| Bit 0 = 1 | The measurement of the total runtime for all part programs is active ($AC_OPERATING_TIME). |
| Bit 1 = 1 | The measurement of the actual program runtime is active ($AC_CYCLE_TIME). |
| Bit 4 = 1 | Measurement also during active dry run feedrate. |
| Bit 5 = 1 | Measurement also during program test. |

## 9.4 Clamping a blank

The blank clamping function allows you to select and specify in more detail the clamping location for the blank.

You require this function if your machine satisfies one or more of the following requirements:

## Requirements

- The machine has several clamping options available for selection.
- The machine has a clamping option that does not point in the Z direction as standard (e.g. a boring mill).

Under the following circumstances, you can specify the position and orientation of the rotary axes in the machine data MD53220 and MD52207:

- The position and orientation of the rotary axis is offset against the MCS zero point, but not specified by a tool holder or a TRAORI.
- In a turning machine, the reference points of the main spindle or counterspindle are offset against the MCS zero point in X or Z.
- The counterspindle cannot be traversed with a saddle, but rather is at a fixed position.

In order that for simulation / simultaneous recording, the rotary axes are displayed at the correct location, their positions in the machine coordinate system must be specified using the following machine data:

| | |
|---|---|
| MD53220 $MAS_AXIS_MCS_POSITION[n] | Position of the axis in the MCS<br>[n] 0 = X, 1 = Y, 2 = Z |

For further information about the transformation of a rotary axis, see Chapter"Technologies and cycles (Page 349)".

## Function

Set the following channel-specific machine data:

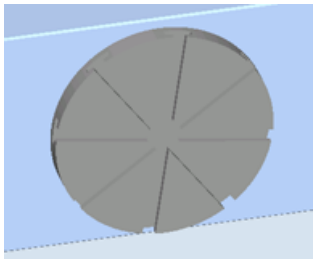
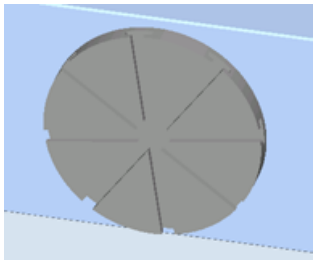
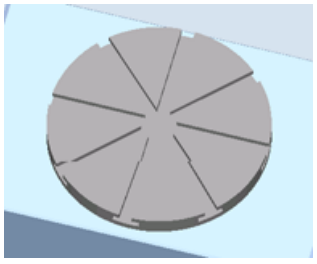
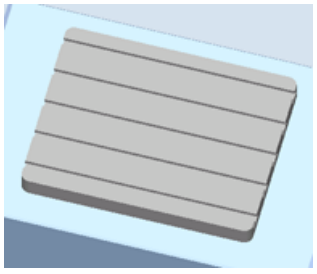
| MD52207 $MCS_AXIS_USAGE_ATTRIB[n] | Attribute of the axes in the channel<br>[n] channel axis number |
|---|---|
| Bit 0 | Rotates around 1st Geometry axis (for rotation axes) |
| Bit 1 | Rotates around 2nd Geometry axis (for rotation axes) |
| Bit 2 | Rotates around 3rd Geometry axis (for rotation axes) |
| Bit 8 | Offer rotary axis for blank clamping (on milling machines) |
| Bit 13 | Position of the axis for blank selection from active work offset (with rotary axes) |
| = 0 | When simulation or simultaneous recording are started, the mountable rotary axis is moved to the position from MD53220. |
| = 1 | With a blank command, The mountable rotary axis is moved into the zero point of the currently active work offset. |

| MD52214 $MCS_FUNCTION_MASK_MILL | Milling function screen |
|---|---|
| Bit 1 | Offer fixed table for blank clamping (on milling machines) |

Take the machine configuration for MD20080 from the example.

## Function options

In the program header, a blank can be optionally set to A, B, C and table clamping:

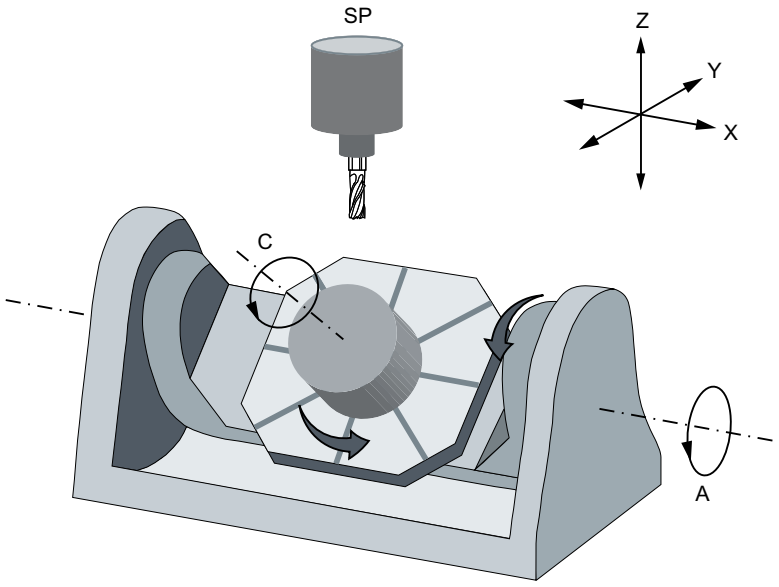
| Clamping | View | Machine data |
|---|---|---|
| A axis |  | MD52207 $MCS_AXIS_USAGE_ATTRIB bit 0 and bit 8 = 1 |
| B axis |  | MD52207 $MCS_AXIS_USAGE_ATTRIB bit 1 and bit 8 = 1 |
| C axis |  | MD52207 $MCS_AXIS_USAGE_ATTRIB bit 2 and bit 8 = 1 |
| Table |  | MD52214 $MCS_FUNCTION_MASK_MILL bit 1 = 1 |

**Note**

To obtain the selection option in the program header, two rotary axes must be defined for the raw part clamping.

## Example for the machine configuration

Example for a machine configuration when the A axis should rotate at the X axis:




| MD20080 $MC_AXCONF_CHANAX_NAME_TAB[n] | Channel axis name in channel [n] |
|---|---|
| [0] | X |
| [1] | Y |
| [2] | Z |
| [3] | Spindle |
| [4] | A |
| [5] | C |

| MD52207 $MCS_AXIS_USAGE_ATTRIB[4] | Attribute of rotary axis A |
|---|---|
| = 181H | |
| Bit 0 = 1 | Rotates around 1st Geometry axis (for rotation axes) |
| Bit 8 = 1 | Offer rotary axis for blank clamping (on milling machines) |

## References

Operating Manual, Milling

- Chapter "Create ShopMill program" > "Program header"
- Chapter "Create ShopMill program" > "Change program settings"

# Spindle functions 10

## 10.1 Spindle control

You can select from the following spindle control settings:

### End of program

The following M functions are used for the end of program:

| | | |
|---|---|---|
| M function from MD10714 $MN_M_NO_FCT_EOP | Main program end | Spindle continues to run |
| M2, M30 | End of main program and return jump to beginning of program | Spindle stops |
| M17 | End of subprogram and return jump into the main program | Spindle continues to run |

If machine data MD10714 $MN_M_NO_FCT_EOP is not equal to zero, then a distinction is made between M2 / M30 (program end of an NC program) and the M function set in MD10714 $MN_M_NO_FCT_EOP.

### Example

MD10714 $MN_M_NO_FCT_EOP = 32 means "M32" for the program end of a program generated in the "JOG" or "MDA" operating mode.

Among other things, this functionality is required in order to continuously start the spindle in manual operation (e.g. for scratching).

### Configuring keys

If you implement a manual control using the keys on the machine control panel, then this is realized via the following interface signals in the spindle data block:

| | |
|---|---|
| DB3x.DBX30.0 | Spindle stop |
| DB3x.DBX30.1 | Spindle start, clockwise rotation |
| DB3x.DBX30.2 | Spindle start, counter-clockwise rotation |

The spindle can be started and stopped in the following state:

| | |
|---|---|
| DB21.DBX35.7= 1 | Channel in reset state |
| DB21.DBX35.6 = 1 | Channel state interrupted |
| DB21.DBX35.3 = 1 | Program state interrupted |

**Note**

If the spindle must be stopped when the program is running, then set the interface signal DB3x.DBX4.3, "Feed stop/spindle stop" in the user PLC.

Additional settings:

| MD11450 $MN_SEARCH_RUN_MODE | | Search run parameterization |
|---|---|---|
| Bit 1 = 1 | Automatic ASUB start after output of the action blocks (see also MD11620 $MN_PROG_EVENT_NAME). Alarm 10208 is not output until the ASUB is completed. | |

### References

For a detailed description how to to configure the spindle, see

Function Manual, Basic Functions; Spindles (S1)

### Direction of rotation (only for ShopTurn)

In the ShopTurn user interface, ensure that the direction of rotation of the spindle and C axis is correctly displayed and when programming ShopTurn functions, that the correct direction of rotation is executed. You must base these settings on the actual direction of rotation of the spindle / C axis on the machine.

- If you use the functions cylinder surface transformation and face machining, then a precondition is that the function was correctly commissioned.
  See Cylinder surface transformation (TRACYL) (Page 390)
- The direction of rotation of the spindle / C axis for the M function M3 that is displayed in the ShopTurn interface is defined in part using machine data:
  See: Defining the direction of rotation in Section: Setting-up ShopTurn cycles for turning (Page 379)
- The direction of spindle rotation (M3 / M4) is assigned to the positive rotational direction of the C axis via interface signal DB31, ... DBX17.6. The bit defines whether M3 and C+ rotate in the same direction (= 0) or in opposite directions (= 1). The corresponding setting options can be found in the following sections:
  - Direction of rotation of counter-spindle Setting up the counterspindle under ShopTurn (Page 384)
  - Direction of rotation of main spindle Setting-up ShopTurn cycles for turning (Page 379)

## 10.2 Analog spindle

### Machine axis index for analog spindle

For a spindle which is not assigned to any drive (FDD, MSD), the utilization display can be controlled via a PLC.

In order that the operating software identifies the spindle as analog spindle, enter the axis index of the analog spindle in the following machine data:

| MD51068 $MNS_SPIND_DRIVELOAD_FROM_PLC1 | Machine axis index spindle 1<br>Utilization display from the PLC |
|---|---|
| Machine axis index of a spindle (analog) that draws the data for the utilization display in the T,F,S window from the PLC (DB19.DBB6). | |

| MD51069 $MNS_SPIND_DRIVELOAD_FROM_PLC2 | Machine axis index spindle 2<br>Utilization display from the PLC |
|---|---|
| Machine axis index of a spindle (analog) that draws the data for the utilization display in the T,F,S window from the PLC (DB19.DBB7). | |

# 10.3 Spindle diagnostics

## 10.3.1 Spindle diagnostics



**Software option**

For the "Spindle diagnostics" you require the "S-Monitor" option.

Main spindle drives are monitored via various status signals. The DRIVE-CLiQ connection at the spindle can be used to evaluate these signals in the drive. The status signals are evaluated for more effective diagnostics of the spindle operating states. The following information is displayed on the user interface with the aid of the "S-Monitor" software option:

- Operating hours
- Temperatures
- Speed/torque
- Clamping system
- Logistics data

### Requirements

- A spindle with DQI encoder: r0459, bit 14 = 1.
- Drive telegram 139 is configured for the spindle.
- The spindle functionality for the machine axis is present when MD35000 $MA_SPIND_ASSIGN_TO_MACHAX > 0
  The value corresponds to the spindle number.

### References

For further information, please refer to the following documentation:

Basic Functions Function Manual: Spindles (S1)
Section "Spindle with SMI 24 (Weiss spindle)"

## Data overview

The "S1 Spindle Diagnostics" window shows the following information:

| Parameter | Value |
|---|---|
| Operating hours<br>• Spindle under control<br>• Spindle under speed | h |
| Number of clamping cycles (tool change)<br>Definition of a clamping cycle: Released clamping state [3] → Clamped clamping state [7, 8 or 10] → Released clamping state [3]. | |
| Motor temperature | °C |
| Clamping state (sensor 1) | V |
| Piston free (sensor 4) | |
| E.g. defined rotary angular position of the shaft (sensor 5) | Yes/No |
| Maximum speed | rpm |

## Procedure

1. Select the "Diagnostics" operating area.
2. Press the menu forward key.
3. Press the "Spindle diagostics" softkey to obtain a data overview.
   The "S1 Spindle Diagnostics" window opens.

**Note**

If several spindles are in operation, select the desired spindle using the "Spindle +" or "Spindle -" softkey.

## See also

Tool change with hybrid spindle from the company Weiss (https://support.industry.siemens.com/cs/ww/en/view/109475707)

## 10.3.2 Temperatures

Which temperatures are evaluated and displayed depends on the number and the mounting location (e.g. for sensor S6, bearing temperature front) of the sensors installed in the spindle. The motor temperature is not displayed when the temperature sensor sensor type parameter p0601 = 0 or 1.

**Note**

**Temperature limits**

The limit values displayed here refer to the spindle monitoring and have no effect on the drive parameter values set in the drive that apply for the shutdown response of the spindle. The temperature limit values are preset in the SMI24 and cannot be changed in this window. If necessary, you must adapt the shutdown limit values for the drive.

Up to three temperatures can be evaluated:

| Signal | Sensor | Column/meaning | Description |
|---|---|---|---|
| Motor temperature (Page 155) | KTY | Actual | Current motor temperature. If an overtemperature protection is present and responds, the value 250° C is displayed. |
| Motor temperature<br>Alarm | | Limit | Stored threshold for motor overtemperature as of which an alarm is issued due to KTY motor overtemperature. |
| | | Duration of the upper limit violation | Displays the accumulated hours during which an upper limit violation was present. |
| | | Last upper limit violation | Displays the time stamp [date] and [time] at which the last upper limit violation was present. |
| | | Number of<br>upper limit violations | Accumulated number of upper limit violations. |
| Motor temperature<br>Fault | | Limit | Stored threshold for motor overtemperature as of which a fault is issued due to KTY motor overtemperature. |
| | | Duration of the upper limit violation | Displays the accumulated hours during which an upper limit violation was present. |
| | | Last upper limit violation | Displays the time stamp [date] and [time] at which the last upper limit violation was present. |
| | | Number of<br>upper limit violations | Accumulated number of upper limit violations. |

| Signal | Sensor | Column/meaning | Description |
|---|---|---|---|
| Overtemperature protection | PTC | Actual | When the overtemperature protection responds, "Active" is displayed (in red). The value 250° C is displayed in the "Actual" column as motor temperature. |
| | | Limit | Stored threshold for motor overtemperature as of which a fault is issued due to PTC winding motor overtemperature. |
| | | Last upper limit violation | Displays the time stamp [date] and [time] at which the last upper limit violation was present. |
| | | Number of upper limit violations | Accumulated number of upper limit violations. |
| Additional temperature (Page 155) (e.g. bearing temperature front) | S6 (KTY) | Actual | Current temperature at the measuring point. |
| Additional temperature Alarm | | Limit | Stored temperature threshold as of which an alarm is issued, e.g. due to bearing overtemperature. |
| | | Duration of the upper limit violation | Displays the accumulated hours during which an upper limit violation was present. |
| | | Last upper limit violation | Displays the time stamp [date] and [time] at which the last upper limit violation was present. |
| | | Number of upper limit violations | Accumulated number of upper limit violations. |
| Additional temperature Fault | | Limit | Stored temperature threshold as of which a fault is issued, e.g. due to bearing overtemperature. |
| | | Duration of the upper limit violation | Displays the accumulated hours during which an upper limit violation was present. |
| | | Last upper limit violation | Displays the time stamp [date] and [time] at which the last upper limit violation was present. |
| | | Number of upper limit violations | Accumulated number of upper limit violations. |

## 10.3.3 Motor temperature sensor

### Shutdown response

The alarm and fault thresholds of the motor temperature sensor that are displayed on the user interface as limits, are stored in the SMI24. They are used for the statistical evaluation of the SMI24 and for the shutdown response of the drive. A delay time setting for the violation of the alarm threshold stored in the SMI24 until the fault is triggered is not relevant for the display.

Please note: The motor parameter values of p0604, p0605 (alarm/fault threshold) and the delay time p0606 are predefined and cannot be changed.

The delay time in p0606 has no effect on the values displayed on the user interface. This means: When the alarm threshold is violated and the delay time set in p0606 has expired, the "Motor temperature fault" signal does not change the "Last upper limit violation" display value and the "Number of upper limit violations" is not incremented.

The parameter values of p0604, p0605 (alarm/fault threshold) and p0606 (delay time) have an effect on the shutdown response of the spindle. The spindle is shut down when the fault threshold is violated or the alarm threshold is violated and the delay time set in p0603 has expired.

Currently, the delay time is not predefined by the SMI24. The parameter value of p0603 is set to the factory setting (240 s).

Alarms:

- 207015 = motor temperature sensor alarm
- 207016 = motor temperature sensor fault

## 10.3.4 Additional temperature sensor

### Shutdown response

The alarm and fault thresholds of the additional temperature sensor that are displayed on the user interface as limits, are stored in the SMI24. They are used only for the statistical evaluation of the SMI24. A delay time setting for the violation of the alarm threshold stored in the SMI24 until the fault is triggered is not relevant for the display.

Please note: The parameter values of p4102[0..1] (alarm/fault threshold) and p4103 (delay time) are predefined and cannot be changed.

Changing the parameter values of p4102[0..1] (alarm/fault threshold) and p4103 (delay time) has no effect on the values displayed on the user interface.

The parameter values of p4102[0..1] (alarm/fault threshold) and p4103 (delay time) have an effect on the shutdown response of the spindle. The spindle is shut down when the fault threshold is violated or the alarm threshold is violated and the delay time set in p4103 has expired. These parameters can be adapted accordingly to prevent an unwanted shutdown of the spindle.

Currently, the delay time is not predefined by the SMI24. The parameter value of p4103 is set to the factory setting (0 s). This means: The timer is deactivated.

Alarms:

- 207017 = additional temperature alarm threshold violated
- 207018 = additional temperature fault threshold violated

## 10.3.5 Temperature histograms

The motor temperature and optionally additional temperatures, such as bearing temperature or housing temperature, are displayed in a histogram. The color coding indicates the following temperature ranges:

| Color | Meaning |
|---|---|
|  | Temperature range up to the first limit. |
|  | Temperature range between the first and second limit. |
|  | Temperature range above the second limit. |

The heights of the individual bar fields correspond to the parameterized temperature ranges of the relevant temperature measuring point.

The specified times for the individual temperature ranges of a measuring point (sensor) are the operating hours under speed.

The currently measured temperatures are displayed below the bars.

## 10.3.6 Speed/torque

To check the loads on the spindle, the speed and the torque are recorded in a histogram as a function of the operating hours. The color coding indicates the following states:

- Light green: low load
- Green: rated load
- Yellow: high load
- Orange: critical load

The heights of the individual bars correspond to the percentage time shares (dwell time) in relation to the operating hours under speed. The accumulated number of operating hours under speed for the spindle are shown above or below the respective bar for the corresponding speed or torque range. The width of the bar corresponds to the evaluated speed or torque range.

**Possible conclusions**

- The speed histogram is an indication of the load on the bearings and rotary gland caused by the speed.
- The speed histogram is an indication of the load on the bearings caused by the stock removal forces. The relationship between torque and radial force changes depending on the tool diameter (lever arm), the milling arc (superimposition of the cutting forces of the individual cutting edges) and the cutting force coefficients (the stock removal force consists of cutting force, feedrate force and passive force - only the cutting force is acquired via the torque).

## 10.3.7 Clamping system

The diagnosis of the clamping system offers the following capabilities:

- Fast and robust detection of the clamping states.
- Monitoring of the clamping task and the clamping system.
- Sensors S1 and S4 in the spindle provide information about the clamping device.
- Sensor S5 detects a position, e.g. the defined rotatory angular position of the shaft.

**Clamping states**

The preset limits for the three clamping states are displayed in the upper section of the window.

The clamping state is defined for the specified voltage range of the analog output signal of sensor S1.

The values in the "Voltage min." and "Voltage max." columns indicate in which voltage range the analog signal of sensor S1 must lie for the corresponding clamping state.

The preset speed limitation "Speed max." specifies the maximum permissible speed for the clamping state and is limited to this by the drive.

**Signal overview**

The current state values of the sensors, the tool clamping, the speed setpoint and the active speed limitation are displayed in the signal overview

| Signal | Sensor | Description | Value | Meaning |
|---|---|---|---|---|
| State | S1 | Clamping state | 3 | Released |
| | | | 7 or 8 | Clamped with tool |
| | | | 10 | Clamped without tool |
| Piston free | S4 | Plunger sensing | Yes | Release plunger not in contact with the shaft. |
| | | | No | Release plunger in contact with the shaft. |
| Shaft in the change position | S5 | Position sensing | Yes | E.g. shaft is at a defined rotary angular position. |
| | | | No | E.g. shaft is not at a defined rotary angular position. |
| Speed setpoint | | Specification of the speed | | Speed specification by the user. |

| Signal | Sensor | Description | Value | Meaning |
|---|---|---|---|---|
| State | | Clamping states | 0 | Sensor S1 not available or state values inactive. |
| | | | 1 | The state initialization is in progress. |
| | | | 2 | Released with signal (error state) |
| | | | 3 | Released |
| | | | 4 | Clamping with tool |
| | | | 5 | Releasing with tool |
| | | | 6 | Releasing without tool |
| | | | 7 | Clamped with tool and S4 = 0 (No). |
| | | | 8 | Clamped with tool and S4 = 1 (Yes). |
| | | | 9 | Clamping without tool |
| | | | 10 | Clamped without tool |
| | | | 11 | Clamped with signal (error state) |

**State bar diagram**

The clamping state is displayed by a state bar. The ranges shown in green indicate the state ranges for "Clamped with or without tool" and "Tool released". The arrow marking indicates the current analog sensor voltage of sensor S1.

For information on the associated interface signals (NC → PLC), refer to the following documentation:

## References

NC Variable and Interface Signals List Manual, Section "Interface signals - Overview" > "Axis/spindle-specific signals".

## Procedure

1. The "S1 Spindle Diagnostics" window is open.
   The "S1 Clamping System" window is open without the "S-Monitor" software option.
2. Press the "Clamping system" softkey.
   The "S1 Clamping System" window opens and displays the acquired data.
3. Press the "Statistics" softkey to read out the clamping times and perform a diagnosis of the clamping system.
   The "S1 Clamping System Statistics" window opens and displays the acquired data.
   - OR -
   Press the "Back" softkey to return to the data overview.

## 10.3.8 Clamping system: Speed limits

For safety reasons and to protect the spindle, only maximum speeds are permitted for certain clamping states. The drive limits the spindle speed in the respective clamping state to the entered speed.

| Clamping state | Meaning |
|---|---|
| Released | The tool clamping system is in contact with the release plunger. Please note: The release plunger presses with the release force on the clamping system. |
| Clamping | The tool clamping system is in the transition state from "Released" to "Clamped with tool". |
| Releasing from state "Clamped with tool" | The tool clamping system is in the transition state from "Clamped with tool" to "Released". |
| Releasing from state "Clamped without tool" | The tool clamping system is in the transition state from "Clamped without tool" to "Released". |
| Clamped with tool | The tool clamping system is in the "Clamped with tool" state. |
| Clamping without tool | The tool clamping system is in the transition state from "Clamped with tool" to "Clamped without tool". |
| Clamped without tool | The tool clamping system is in the "Clamped without tool" state. |

You can adapt the preset speed limits to your requirements by pressing the "Change" softkey.

| NOTICE |
|---|
| **Damage to the spindle through too high a speed**<br>Changes to the preset speed limits can result in changes of the operating states and to damage on the spindle. |

### Procedure

1. The "S1 Spindle Diagnostics" window is open.
   The "S1 Clamping System" window is open without the "S-Monitor" software option.
2. Press the "Clamping system" softkey.
   The "S1 Clamping System" window opens and displays the acquired data.
3. Press the "Speed limitation" softkey to change the limit values for the speeds of the clamping states.
   The "S1 Speed Limitations" window opens.
4. Press the "Change" softkey to enter the desired speed limits depending on the clamping state.
   Confirm with "OK".

## 10.3.9 Clamping system: Diagnostic statistics

This window displays the statistics for clamping operations and speed violations.

| Statistical data | Meaning |
|---|---|
| Number of correct clamping cycles | Accumulated number of correct clamping cycles with or without tool.<br>Correct clamping cycle means: "Released" clamping state [3] → "Clamped" clamping state [7, 8 or 10] → "Released" clamping state [3]. A clamping cycle is considered to be correct when a clamping and subsequent release operation are performed via the appropriate clamping states (without "State return"). Starting from one clamping state, the next clamping state that can be reached must not be the previous state. There is no dependency on time. |
| Clamping time (without tool) | The time within a clamping operation between the clamping states "Tool released" and "Clamped without tool". |
| • Within tolerance | Accumulated number of clamping operations without tool that were within the specified time. |
| • Outside tolerance | Accumulated number of clamping operations without tool that were outside the specified time. |
| • Specification | Factory specification of the time for the clamping operation without tool. |
| • Current | Measured time for the last performed clamping operation without tool. |
| Clamped (without tool) | Diagnostics during the "Clamped without tool" clamping state. |
| • Speed violations | Accumulated number of events during which the spindle speed was greater than the set limit as delivered. |
| • Speed threshold | Limit for the spindle speed in the "Clamped without tool" clamping state (as-delivered value).<br>Changing this speed limit via the diagnostics overview "S1 clamping system" → "Speed limits" → "Change" has no effect on the value specified here. |
| Faulty clamping/releasing operations (total) | Accumulated number of clamping operations that have not run through the clamping states "Released" clamping state [3] → "Clamped" clamping state [7, 8 or 10] → "Released" clamping state [3] without interruption.<br>A clamping cycle is considered to be faulty when, starting from one state, the next state that is reached is the previous state again ("State return"). |

## 10.3.10 Clamping system: Clamping time statistics

This window displays the statistics for the determined clamping times.

| Statistical data | Meaning |
|---|---|
| Last determined clamping time | Measured time for the last performed clamping operation.<br>Definition of a clamping operation: "Released" clamping state [3] → "Clamped with tool" clamping state [7 or 8]. Clamping state 3 = Released; clamping state 7 = Clamped with tool and S4 = 0 (No); clamping state 8 = Clamped with tool and S4 = 1 (Yes) |
| Reference clamping time (with tool) | During the spindle start-up phase, the first 100 clamping operations are measured by the SMI24 and the minimum value, maximum value and the mean value determined. |
| • Minimum | The shortest clamping time determined from the first maximum 100 clamping operations. |
| • Maximum | The longest clamping time determined from the first maximum 100 clamping operations. |
| • Mean | The mean clamping time determined from the first maximum 100 clamping operations. |
| Mean clamping time (without tool) | After the first 100 clamping times determined for the reference time, the mean values for the other measured clamping times are calculated. |
| • 24 h | Mean clamping time of the clamping cycles determined in the last 24 hours. |
| • 10 days | Mean clamping time of the clamping cycles determined in the last 10 days. |
| • 100 days | Mean clamping time of the clamping cycles determined in the last 100 days. |

If a Power On of the SMI24 is performed during the start-up phase, the previously determined values are discarded. The counter for the first 100 clamping operations is set to zero and the reference times up to the 100th clamping operation can be determined again. After the start-up phase (first 100 clamping operations), the values of the reference clamping time are "frozen". Even after a Power On of the SMI24, the values remain stored and can no longer be changed.

## 10.3.11 Fetching the logistics data

### Requirement

The spindle characteristics can only be read out with the "Service" access level.

### Spindle characteristics

The following spindle manufacturer characteristics can be read out via DRIVE-CLiQ:

- Manufacturer's identifier
- Drawing number

- Serial number
- Date of manufacture
- Service date and service information (1...4):
  The service information corresponds to the reference number for the service report from the WEISS Spindeltechnologie GmbH company.

# Drive system

# 11

## 11.1 Commissioning of drives

### Procedure

Once commissioning of the PLC has been completed, you can carry out commissioning of the SINAMICS drives via the user interface of SINUMERIK Operate.




1. Select the "Start-up" operating area.




2. Press the "Drive system" softkey.
   The "Drive system overview" window opens.
   A message is output whether the drive system has already been commissioned or not.

### References

The procedure for commissioning the SINAMICS drives is described in the following documentation:

CNC Commissioning Manual: NCK, PLC, drive

# Configuring alarms

# 12

## 12.1 Creating alarm and message texts via the user interface

From the user interface, you can create and edit your alarm and message texts from the part program. In turn, the alarm and message texts are saved corresponding to their number in various text files.

### Storage

The text files are created in the directory:
/**oem**/sinumerik/hmi/lng

### Procedure




1. Select the "Start-up" operating area.




2. Press the "HMI" softkey.




3. Press the "Alarm texts" softkey.
   The "Select File" window opens and offers the user-specific text files.




4. Select the desired file and press the "OK" softkey.

5. Enter the desired alarm number in the "Number" column.
   Enter the desired alarm text in the "Alarm text" column.
   Select the desired font color in the "Color" column.
   In the column "PopUp" with "yes" or "no", select whether the alarm text should be displayed in a window that requires acknowledgment - or not.
   - AND / OR -




   Press the "Insert line" softkey.
   A new line is inserted above your cursor.
   - OR




   Press the "Delete line" softkey to delete the selected line.




6. Press the "Search" softkey.
   The "Search" window opens.
   Enter the desired alarm text or the alarm number in the "Text" field.

Activate the "Case sensitive" checkbox if a distinction must be made between upper- and lower-case letters when entering text.

- OR

Search + replace

Press the "Search and replace" softkey.

The "Search and Replace" window opens.

Enter the search term in the "Text" field. Enter the replacing term in the "Replace with" field.




7. Position the cursor in the "Direction" field. Using the <SELECT> key, select the search direction (forwards, backwards).




8. Press the "OK" softkey to start search or search and replace.




Press the "Cancel" softkey if you wish to interrupt the action.




9. If you have entered the alarms with the help texts, press the "OK" softkey.

You will receive the message "The alarm texts have been saved and converted."

## Restart SINUMERIK Operate

You must restart SINUMERIK Operate in order that the "color" and "PopUp" attributes take effect. Modified alarm texts, however, take effect immediately.

## Further search options

Go to start

The cursor jumps to the first entry of the selected alarm text file.

Go to end

The cursor jumps to the last entry of the selected alarm text file.

## Creating search directories

In order that a search can be made for alarm files in subdirectories other than the existing /lng directory, you have the following option:

1. Copy the "systemconfiguration.ini" configuration file from the following directory:

   /**siemens**/sinumerik/hmi/cfg
2. Store the file in the following directory:

   /**oem**/sinumerik/hmi/cfg
3. Create any subdirectory under /hmi, e.g. /oem_dir.

4. Create the following section in the configuration file "systemconfiguration.ini":
```
[oem_dirs]
OEM_3=oem_dir
```
5. Restart SINUMERIK Operate.
   When searching, the subdirectory is now also taking into account.

### Foreign language alarm and message texts

If you create foreign language alarm and message texts, change over into the desired language. All text files are automatically allocated the language code corresponding to the user interface language that has been selected. For instance, if you create alarm texts in the English user interface, then the text file is assigned the "eng" language code.

### See also

Range of alarms (Page 184)

## 12.2 Configuring alarm and message texts via alarm text files

You have the option of creating customized alarm and message texts using alarm text files and adapting these.

The alarm texts are set up in standard format (".ts" format) by SINUMERIK Operate. The ".ts" format is XML-based.

You can create and edit these files not only with SINUMERIK Operate, but also externally on a PC.

**Note**

If you wish to edit alarm text files on a PC, use an editor supporting UTF-8 coding.

**Note**

Chinese alarm texts should only be implemented with character set GB2312.

Chinese logograms are not supported in Korean.

### Creating and editing alarm texts

- Creating your own alarm texts
- Creating texts for indexed alarm parameters
- Creating part program message texts
- Changing alarm attributes

- Replacing standard alarm texts
- Converting alarm texts

## 12.2.1 Creating in-house alarm texts

### Creating an alarm text file

1. You can copy a sample file "oem_alarms_eng.ts" from the following directory: /**siemens**/sinumerik/hmi/template/lng
2. Save or create the file in the /**oem**/sinumerik/hmi/lng or /**user**/sinumerik/hmi/lng directory. Create a separate file for each country language that is supported.
3. Assign a unique name to the file, e.g. "my_alarms_eng.ts". You can freely select the file names with the exception of the language code and the extension.
   The file name always ends with the language code of the particular language and always has the ".ts" extension. For example "my_alarms_deu.ts" for German or "my_alarms_eng.ts" for English.

### Creating alarm texts

1. Open the "my_alarms_eng.ts" file.
2. For each alarm text, you must insert your own area in the file that is marked using the <message> tag.
3. The <name> tag contains the context names. Normally, you should not change the specified name "`slaeconv`".
4. The <source> tag contains the alarm number and the name of the alarm source (source URL). In the example "700000" is the alarm number and "/PLC/PMC" is the name of the alarm source. The possible alarm sources can be found in the tables in Section Range of alarms (Page 184).
5. The <translation> tag contains the alarm text.

### Example:

```
<!DOCTYPE TS>
<TS>
    <context>
       <name>slaeconv</name>
       <message>
            <source>700000/PLC/PMC</source>
            <translation>First OEM alarm text</translation>
       </message>
       <message>
```

```
            <source>700001/PLC/PMC</source>
            <translation>Second OEM alarm text</translation>
        </message>
    </context>
</TS>
```

### Registering the alarm text file

You still have to register your alarm text file so that the system knows your alarm texts during the program runtime:

1. You can copy a sample configuration file "oem_slaesvcadapconf.xml" from the following directory: /**siemens**/sinumerik/hmi/template/cfg
2. Save the file in the directory /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg
3. Assign the name "slaesvcadapconf.xml" to the file.
4. Open the file and in the <BaseNames> tag, enter the file name of the alarm text file without language code and file extension, e.g. "my_alarms", as "value".
   The registration of your alarm text file starts, for example, with the name "OEM_BaseName_01".

   **Note**

   The names of the <BaseNames> tag can be freely selected. They must be unique in the system and must not clash with names used by Siemens.

   The "Siemens_BaseName_01" name is reserved for Siemens.

   Several alarm text files can also be registered. To do this, use names such as "OEM_BaseName_02", "OEM_BaseName_03", etc.

### Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Configuration of the Solutionline Alarm & Event Service Adapter
-->
<CONFIGURATION>
    <AlarmTexts>
        <BaseNames>
            <OEM_BaseName_01 type="QString" value="my_alarms"/>
        </BaseNames>
    </AlarmTexts>
    ...
</CONFIGURATION>
```

### Restart SINUMERIK Operate

The files must still be converted into a binary format so that the alarm texts can be displayed during the program runtime. This data is only converted during power up.

To do this, restart SINUMERIK Operate. In the same directory where the .ts files are located, files with the same name are created with the ".qm" file extension.

The result of the conversion is written to the "alarmtext_conversion.log" or "oem_text_conversion.log" file. Errors that occur during the conversion, such as syntax errors in a parameter file, are also written to the file.

This file is located in the "/user/sinumerik/hmi/log" directory.

**Note**

The conversion is only made if the .ts file is newer than the associated .qm file.

## 12.2.2 Creating texts for indexed alarm parameters

If a string, the so-called "Index identifier", is specified in the alarm text in pointed brackets after a space holder (e.g. "%1<OEM>"), the parameter itself is not inserted in the alarm text, but a text from another alarm text file, a so-called index text file. In this case, the alarm parameter value serves as index in order to select the text in the index file. The index text can, in turn, include parameters, also index parameters.

**Note**

**Explanation for the "Index identifier" string**

- "<" ≙ opening pointed bracket "<"
- ">" ≙ closing pointed bracket ">"

### Creating an index text file

1. You can copy a sample index text file "oem_indexparams_eng.ts" from the following directory: /**siemens**/sinumerik/hmi/template/lng
2. Save or create the file in the /**oem**/sinumerik/hmi/lng or /**user**/sinumerik/hmi/lng directory. Create a separate file for each country language that is supported.
3. Assign a unique name to the file, e.g. "my_indextexts_eng.ts". You can freely select the file names with the exception of the language code and the extension.
   The file name always ends with the language code of the particular language and always has the ".ts" extension. For instance, "my_indextexts_deu.ts" for German or "my_indextexts_eng.ts" for English.

### Creating index texts

1. Open the "my_indextexts_eng.ts" file.
2. In the <name> tag, enter a name that you have selected, e.g. "my_context".
3. For each index text, a separate area must be inserted between the <message> and </message> tags.
4. The <source> tag contains the alarm parameter value, e.g. the values "1" and "2".
5. The <translation> tag includes the index text that is displayed if the alarm parameter involved has the value specified between <source> and </source>.

### Example

```
<!DOCTYPE TS>
<TS>
    <context>
       <name>my_context</name>
       <message>
            <source>1</source>
            <translation>First OEM parameter text</translation
       </message>
       <message>
            <source>2</source>
            <translation>Second OEM parameter text</translation
       </message>
    </context>
</TS>
```

### Registering an index text file

You still have to register your index text file so that the system knows your index texts during the program runtime. The registration is made in the "slaesvcadapconf.xml" file.

1. Open the file "slaesvcadapconf.xml" already created for the alarm texts /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg directory.
2. Remove the lines "<!-- Start of comment" and "End of comment -->".
3. Enter the so-called identifier, e.g. <Identifier type="QString" value="OEM"/>. The index identifier is always specified in pointed brackets next to the parameter specification in the alarm text, e.g. "%1<OEM>".
4. Enter the BaseName of the index text file, e.g. <BaseName type="QString" value="my_indextexts"/>.

5. Enter the context name that you selected, e.g.
   <ContextName type="QString" value="my_context"/>.
6. You can also use several different indices. Then, in this case, a dedicated section must be created for every index between the <IndexTexts> and </IndexTexts> tags. The tags for these sections are called, e.g.
   <OEM_IndexText_01>, <OEM_IndexText_02>, <OEM_IndexText_03>, etc.
   The tags <IndexText_01> to <IndexText_99> are reserved for Siemens.

## Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!- Configuration of the Solutionline Alarm & Event Service Adapter
->
<CONFIGURATION>
...
    <!-- The following narrated part of the configuration is used
    only when 'Indexparameters' (e.g. '%1&lt;OEM&gt;') are used
    within OEM alarm texts. In this case the value of the parameter
    is used as an index into an additional text list to reference
    another text which is placed into the alarm text instead of the
    original parameter value. -->
    <IndexTexts>
       <OEM_IndexText_01>
          <Identifier type="QString" value="OEM"/>
          <BaseName type="QString" value="my_indextexts"/>
          <ContextName type="QString" value="my_context"/>
          <MetaTextID type="QString" value="%ParamValue%"/>
       </OEM_IndexText_01>
    </IndexTexts>
</CONFIGURATION>
```

## Restart SINUMERIK Operate

The files must still be converted into a binary format so that the alarm texts can be displayed during the program runtime. This data is only converted during power up.

To do this, restart SINUMERIK Operate. In the same directory where the .ts files are located, files with the same name are created with the ".qm" file extension.

The result of the conversion is written to the "alarmtext_conversion.log" or "oem_text_conversion.log" file. Errors that occur during the conversion, such as syntax errors in a parameter file, are also written to the file.

This file is saved in the /user/sinumerik/hmi/log directory.

**Note**

The conversion is only made if the .ts file is newer than the associated .qm file.

## 12.2.3 Creating part program message texts

### Message texts from the part program

In order to output message texts from the part program, use the MSG instruction in the part program. The message texts are referenced via the number after the "$ character, e.g. MSG("$4711"). You can configure channel-specific message texts for each NC channel.

### Creating message texts

Procedure:

1. You can copy a sample message text file "oem_pp_messages_eng.ts" from the following directory: /**siemens**/sinumerik/hmi/template/lng
2. Save or create the file in the /**oem**/sinumerik/hmi/lng or /**user**/sinumerik/hmi/lng directory. Create a separate file for each country language that is supported.
3. Assign a unique name to the file, e.g. "my_msgs_eng.ts". You can freely select the file names with the exception of the language code and the extension.
   The file name always ends with the language code of the particular language and always has the ".ts" extension. For instance, "my_msgs_deu.ts" for German or "my_msgs_eng.ts" for English.
4. Open the "oem_msgs_eng.ts" file and in the <name> tag enter the "partprogmsg01" character string. This is the default for the part program message texts from all channels.
5. For each message text, a separate area must be inserted between the <message> and </message> tags.
6. The <source> tag contains the number from the MSG command in the part program.
7. The <translation> tag contains the message text.

### Example

```
<!DOCTYPE TS>
<TS>
   <context>
     <name>partprogmsg01</name>
     <message>
        <source>4711</source>
        <translation>part program message No. 4711</translation
     </message>
   </context>
</TS>
```

## Registering the message text file

You still have to declare your message text file so that the system knows your message texts during the program runtime.

1. You can copy a sample configuration file "oem_slaesvcadapconf.xml" from the following directory: /**siemens**/sinumerik/hmi/template/cfg
2. Save or create the file in the /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg directory.
3. Change the name to "slaesvcadapconf.xml".
4. Open the file and in the <BaseNames> tag enter the file name of the message text file without language code and file extension, e.g. "my_msgs", as "value". The registration of your message text file starts, for example, with the name "OEM_BaseName_01".

   **Note**

   The names of the <BaseNames> tag can be freely selected. They must be unique in the system and must not clash with names used by Siemens.

   The "Siemens_BaseName_01" name is reserved for Siemens.

   Several message text files can also be registered. To do this, use names such as "OEM_BaseName_02", "OEM_BaseName_03", etc.

## Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Configuration of the Solutionline Alarm & Event Service Adapter
-->
<CONFIGURATION>
    <AlarmTexts>
       <BaseNames>
          <OEM_BaseName_01 type="QString" value="my_msgs"/>
       </BaseNames>
    </AlarmTexts>
    ...
</CONFIGURATION>
```

## Message texts for several channels

It is possible to assign different message texts the same message number, e.g. the number 4711 in different NC channels. This means, different message texts are output depending on the channel in which the part program is executed.

1. Open the "oem_msgs_eng.ts" file in the editor.
2. Change the text between the two <name> and </name> tags from "partprogmsg01" into "partprogmsgXY", whereby "XY" should be replaced by the particular channel number, e.g. "partprogmsg02" for channel 2.

3. Now open the "slaesvcconf.xml" configuration file.
4. Enter the respective NC channel in the <Connections> tag. Only use the XML tags from the following table to specify the NC channel.

| NC channel | XML tag | ContextName |
|---|---|---|
| 1 | PartprogramMessageChannel_01 | partprogmsg01 |
| 2 | PartprogramMessageChannel_02 | partprogmsg02 |
| 3 | PartprogramMessageChannel_03 | partprogmsg03 |
| 4 | PartprogramMessageChannel_04 | partprogmsg04 |
| 5 | PartprogramMessageChannel_05 | partprogmsg05 |
| 6 | PartprogramMessageChannel_06 | partprogmsg06 |
| 7 | PartprogramMessageChannel_07 | partprogmsg07 |
| 8 | PartprogramMessageChannel_08 | partprogmsg08 |
| 9 | PartprogramMessageChannel_09 | partprogmsg09 |
| 10 | PartprogramMessageChannel_10 | Partprogmsg10 |

## Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Configuration of the Solutionline Alarm & Event Service Adapter
-->
<CONFIGURATION>
    <Connections>
       <PartprogramMessageChannel_02>
          <ContextName type="QString" value="partprogmsg02"/>
       </PartprogramMessageChannel_02>
    </Connections>
</CONFIGURATION>
```

## Restart SINUMERIK Operate

The files must still be converted into a binary format so that the alarm texts can be displayed during the program runtime. This data is only converted during power up.

To do this, restart SINUMERIK Operate. In the same directory where the .ts files are located, files with the same name are created with the ".qm" file extension.

The result of the conversion is written to the "alarmtext_conversion.log" or "oem_text_conversion.log" file. Errors that occur during the conversion, such as syntax errors in a parameter file, are also written to the file.

This file is saved in the /user/sinumerik/hmi/log directory.

**Note**

The conversion is only made if the .ts file is newer than the associated .qm file.

## 12.2.4 Changing alarm attributes

### Color types of the alarms

You can individually change the colors of the alarms and messages displayed in the message line.

You can change the following for each alarm number and alarm source:

- Font color of the alarm/message text
- Background color of the alarm/message text
- Font color of the alarm number
- Background color of the alarm number

### Create alarm attribute file

Procedure:

1. You can copy a sample alarm attribute file "oem_slaedatabase.xml" from the directory: /**siemens**/sinumerik/hmi/template/cfg/
2. Save the file in the directory /**oem**/sinumerik/hmi/cfg/ or /**user**/sinumerik/hmi/cfg/
3. Assign a unique name to the file, e.g. " muster_slaedatabase.xml ".
   The file name can be freely selected; however, it must only contain lower-case letters.

### Defining alarm colors

Procedure:

1. Open the file "muster_slaedatabase.xml" in the editor.
2. First, enter which alarm colors you generally wish to change.
   To do this, create a separate section <Attribute> for each alarm color that you wish to change in the <Attributes> area.

3. Enter the attribute name of the alarm color to be changed in the <Attribute AttrName=…> tag. You can change the following colors:
   - Font color of the alarm/message text = TEXTCOLOR
   - Background color of the alarm/message text = TEXTBACKGROUNDCOLOR
   - Font color of the alarm number = NUMBERCOLOR
   - Background color of the alarm number = NUMBERBACKGROUNDCOLOR
4. Alarm sources are defined in the <Sources> area for whose alarms the alarm colors are to be changed, e.g. the alarm sources "/HMI" and "/PLC/PMC".

   **Note**

   If you insert further alarm sources, make sure that the alarm number is always assigned to the correct alarm source.

   The possible values for source ID and URL can be found in the table in Section Range of alarms (Page 184).

5. In the <Alarms> tag, create a separate sub-area for each alarm or for an alarm number range.
6. Enter the alarm number in the <Alarm AlarmID= "..." > tag, or enter the alarm number range in the <Range FromAlarmID="..." ToAlarmID= "..."> tag.
7. Enter the required color values in the following tags:
   <TEXTCOLOR>
   <TEXTBACKGROUNDCOLOR>
   <NUMBERCOLOR>
   <NUMBERBACKCOLOR>
   The attributes of the color values are defined by specifying an RGB value:
   - An RGB value always starts with the "#" character.
   - The individual color values R, G and B are represented by double-digit hexadecimal numbers. Syntax: "#RRGGBB", e.g. "#FF9A00".

## Example

Note the order in the XML configuration file for the assignment of the colors to alarm number ranges:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SlAeAlarmAttributes>
<SlAeAlarmAttributes Version="01.00.00.00">
<Types>
  <Type TypeName="Condition" TypeID="32">
   <Category Version="1.0" CatID="1">
      <CatDescr>Alarms of the SINUMERIK 840D sl.</CatDescr>
      <Attributes>
       <Attribute AttrName="TEXTCOLOR" AttrID="5003"
       AttrDataType="10">
```

```
            <AttrDescr>
              Text color of an alarm used when displayed within the
              header panel.
            </AttrDescr>
          </Attribute>
          <Attribute AttrName="TEXTBACKGROUNDCOLOR" AttrID="5004"
          AttrDataType="10">
            <AttrDescr>
              Background color of an alarm used when displayed within
              the header panel.
            </AttrDescr>
          </Attribute>
          <Attribute AttrName="NUMBERCOLOR" AttrID="5005"
          AttrDataType="10">
            <AttrDescr>
              Text color of an alarm used when displayed within the
              header panel.
            </AttrDescr>
          </Attribute>
          <Attribute AttrName=" NUMBERBACKGROUNDCOLOR " AttrID="5006"
          AttrDataType="10">
            <AttrDescr> Background color of an alarm used when
            displayed within the header panel.
            </AttrDescr>
          </Attribute>
        </Attributes>
      </Category>
    </Type>
</Types>
<Sources>
    <Source> CatLink="1" SourceID="10000" SourceURL="/HMI">
      <Alarms>
        <Alarm AlarmID="130000">
          <TEXTCOLOR>#000000</TEXTCOLOR>
          <TEXTBACKGROUNDCOLOR>#FFFFFF</TEXTBACKGROUNDCOLOR>
          <NUMBERCOLOR>#FFFFFF</NUMBERCOLOR>
          <NUMBERBACKGROUNDCOLOR>#000000</NUMBERBACKGROUNDCOLOR
        </Alarm>
      </Alarms>
    </Source>
<Source> CatLink="1" SourceID="51" SourceURL="/PLC/PMC">
      <Alarms>
        <Range FromAlarmID="700100" ToAlarmID="700199">
          <TEXTCOLOR>#000000</TEXTCOLOR>
```

```
        <NUMBERCOLOR>#00FF00</NUMBERCOLOR>
      </Range>
      <Alarm AlarmID="700000">
        <TEXTCOLOR>#000000</TEXTCOLOR>
        <TEXTBACKGROUNDCOLOR>#FFFFFF</TEXTBACKGROUNDCOLOR>
        <NUMBERCOLOR>#FFFFFF</NUMBERCOLOR>
        <NUMBERBACKGROUNDCOLOR>#000000</NUMBERBACKGROUNDCOLOR
      </Alarm>
    </Alarms>
  </Source>
</Sources>
</SlAeAlarmAttributes>
```

### Registering the alarm attribute file

The alarm attribute file with the color definitions must be registered in order that the alarm colors can be assigned to the alarms during the program runtime:

1. You can copy a sample configuration file "oem_slaesvcconf.xml" from the following directory: /**siemens**/sinumerik/hmi/template/cfg
2. Save or create the file in the /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg directory.
3. Name the file "slaesvcconf.xml".
4. Open the file and in the <DataBases> tag enter, for example, the file name of the alarm attribute file without language code and file extension, e.g. "muster_slaedatabase", as "value". The registration of your alarm attribute file starts, for example, with the name "OEM_DataBase_01".

   **Note**

   The names of the <DataBases> tag can be freely selected. They must be unique in the system and must not clash with names used by Siemens.

   The "Siemens_DataBase_01" name is reserved for Siemens.

   If you register additional alarm attribute files, i.e. you wish to include them in the configuration file, use names such as "OEM_DataBase_02", "OEM_DataBase_03", etc.

### Restart SINUMERIK Operate

The files still have to be converted into a binary format in order that the alarm colors become effective during the program runtime. This data is only converted during power up.

To do this, restart SINUMERIK Operate. A file with the same name is created with the ".hmi" file extension, e.g. "sample_slaedatabase.hmi" in the same directory in which the alarm attribute file is also located.

The result of the conversion is written to the file "oem_ae_database_conversion.log". This file is located in the "/user/sinumerik/hmi/log" directory.

**Note**

The conversion is only carried out if the XML file is newer than the associated hmi file.

## 12.2.5 Replacing standard alarm texts

You can replace alarm texts of the standard SINUMERIK Operate with own alarm texts.

### Creating an alarm text file

1. You can copy a sample alarm text file "oem_alarms_eng.ts" from the following directory: /**siemens**/sinumerik/hmi/template/lng
2. Save or create the file in the /**oem**/sinumerik/hmi/lng or /**user**/sinumerik/hmi/lng directory. Create a separate file for each country language that is supported.
3. Assign the file a user-specific name, e.g. "my_nck_alarms_eng.ts". You can freely select the file names with the exception of the language code and the extension.
   The file name always ends with the language code of the particular language and always has the ".ts" extension. For example "my_nck_alarms_deu.ts" for German or "my_nck_alarms_eng.ts" for English.

### Creating alarm texts

1. Open the "my_nck_alarms_eng.ts" file.
2. A separate area must be inserted for each alarm text, marked by the <message> tag.
3. The <name> tag contains the context names. Change the name and ensure that the name is unique in the system, e.g. "myNckAlarms".

   **Note**

   The name for the <name> tag can always be freely selected. However, the name must not clash with the "`slaeconv`" name used by Siemens.

4. The <source> tag contains the alarm number of the standard alarm, e.g. 10000.
5. The <translation> tag contains the actual alarm text, e.g. "OEM alarm text example for NCK alarm 10.000".

### Example

```
<!DOCTYPE TS>
<TS>
   <context>
```

```
    <name>myNckAlarms</name>
    <message>
      <source>10000/NCK</source>
      <translation> OEM alarm text example for NCK alarm 10.000</
      translation>
    </message>
  </context>
<TS>
```

### Registering the alarm text file

You still have to register your alarm text file so that the system knows your alarm texts during the program runtime:

1. You can copy a sample configuration file "oem_slaesvcadapconf.xml" from the following directory: /**siemens**/sinumerik/hmi/template/cfg
2. Save or create the file in the /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg directory.
3. Assign the name "slaesvcadapconf.xml" to the file.
4. Open the file and in the <BaseNames> tag enter the file name of your alarm text file without language code and file extension, e.g. "my_nck_alarms", as "value". The registration of your alarm text file starts, for example, with the name "Siemens_BaseName_01".

   **Note**

   The names of the <BaseNames> tag can be freely selected. They must be unique in the system and must not clash with names used by Siemens.

   The "Siemens_BaseName_01" name is reserved for Siemens.

   Several alarm text files can also be registered. To do this, use names such as "OEM_BaseName_02", "OEM_BaseName_03", etc.

### Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Configuration of the Solutionline Alarm & Event Service Adapter
-->
<CONFIGURATION>
  <AlarmTexts>
    <BaseNames>
      <OEM_BaseName_01 type="QString" value="my_nck_alarms"/>
  </BaseNames>
  </AlarmTexts>
</CONFIGURATION>
```

## Replacing standard alarm texts

You still have to define which standard alarm texts should be replaced by your alarm texts so that during the program runtime, the original alarm text is replaced by your alarm text.

1. Create a new alarm attribute file (see Changing alarm attributes (Page 176)) in the /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg directory. The file name can be freely selected, e.g. "my_nck_alarms_db.xml".
2. Open the file.
3. The file always comprises the two areas <Types> and <Sources>.
4. The <sources> tag includes the reference, assigned to an alarm source, e.g. "\NCK", to the associated alarm text.
5. Link the contents of the <context> <name> tags with the contents of the <message> <source> tags from your alarm text file.
   - <context> <name> = the context name that you assigned
   - <message> <source> = the number of the alarm whose text is to be replaced.
     E.g. enter the following link in the alarm attribute file: <MSGTEXT>myNckAlarms | 10000</MSGTEXT>
     Make sure that the context name and the alarm number are separated by the pipe character "|".
6. If you include an additional NCK alarm, copy the range from <Alarms> to </Alarms> and adapt the alarm number.
7. If you include an additional alarm from another number range (e.g. PLC alarm), copy the range <Sources> to </Sources> and then adapt the SourceID and SourceURL as well as the alarm number.
   You can take the corresponding SourceID and SourceURL from the table in Section Range of alarms (Page 184).

**Note**

A dedicated link must be created for each standard alarm for which the text is to be replaced.

## Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SlAeAlarmAttributes>
<SlAeAlarmAttributes Version="01.00.00.00">
<Types>
   <Type TypeName="Condition" TypeID="32">
     <Category Version="1.0" CatID="1">
       <CatDescr>Alarms of the Sinumerik 810/840 D(i).</CatDescr>
       <Attributes>
       </Attributes>
     </Category>
```

```
   </Type>
</Types>
<Sources>
   <Source CatLink="1" SourceID="0" SourceURL="/NCK">
     <Alarms>
       <Alarm AlarmID="10000">
          <MSGTEXT>myNckAlarms|10000/NCK</MSGTEXT>
       </Alarm>
     </Alarms>
   </Source>
</Sources>
</SlAeAlarmAttributes>
```

### Registering the alarm attribute file

The alarm attribute file still has to be registered in order that the standard alarm texts are replaced during the program runtime:

1. You can copy a sample configuration file "oem_slaesvcconf.xml" from the following directory: /**siemens**/sinumerik/hmi/template/cfg
2. Save or create the file in the /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg directory.
3. Change the name to "slaesvcconf.xml".
4. Open the file and in the <DataBases> tag enter the name of the alarm attribute file, e.g. <OEM_DataBase_01 type="QString" value="my_nck_alarms_db">

   **Note**

   The names of the <DataBases> tag can be freely selected. They must be unique in the system and must not clash with names used by Siemens.

   The <Siemens_DataBase_01> tag name is reserved for Siemens.

   If you register additional alarm attribute files, i.e. you wish to include them in the configuration file, use the names <OEM_DataBase_02>, <OEM_DataBase_03>, etc.

### Restart SINUMERIK Operate

The files with the alarm texts and the attribute file still have to be converted into a binary format in order that the alarm texts can be displayed during the program runtime. This data is only converted during power up.

To do this, restart SINUMERIK Operate. In the same directory where the .ts files or the alarm attribute file are located, files with the same name are created with the ".qm" and .hmi file extension.

The result of the conversion is written to the "alarmtext_conversion.log" or "oem_text_conversion.log" file. Errors that occur during the conversion, such as syntax errors in a parameter file, are also written to the file.

This file is located in the "/user/sinumerik/hmi/log" directory.

**Note**

The conversion is only made if the .ts file is newer than the associated .qm file.

## 12.2.6 Range of alarms

### Number ranges of alarms

| Numerical range | Description | | SourceID | SourceURL |
|---|---|---|---|---|
| 000 000 ... 009 999 | General alarms | | 0 (standard NCU)<br>100 (1. NCU)<br>...<br>9999 (99. NCU) | /NCK<br>//configuration_spec. Designation>//NCK |
| 010 000 ... 019 999 | Channel alarms | | | |
| 020 000 ... 029 999 | Axis/spindle alarms | | | |
| 030 000 ... 039 999 | Functional alarms | General | | |
| 040 000 ... 059 999 | | Reserved | | |
| 060 000 ... 064 999 | | Siemens cycle alarms | | |
| 065 000 ... 069 999 | | Cycle alarms user | | |
| 070 000 ... 079 999 | | Compile cycles<br>Manufacturer and OEM | | |
| 080 000 ... 084 999 | | Siemens cycles message texts | | |
| 085 000 ... 089 999 | | User cycles message texts | | |
| 090 000 ... 099 999 | | Reserved | | |
| 100 000 ... 129 999 | System | | 10 000 | /HMI |
| 130 000 ... 139 999 | OEM | | | |
| 140 000 ... 199 999 | Reserved | | | |
| 200 000 ... 299 999 | SINAMICS drive | | 0 | /NCK |
| 300 000 ... 399 999 | Drive and I/O alarms | | | |
| 400 000 ... 499 999 | General alarms | | 51 | /PLC/PMC |
| 500 000 ... 599 999 | Channel alarms | | | |
| 600 000 ... 699 000 | Axis/spindle alarms | | | |
| 700 000 ... 799 999 | User area | | | |
| 800 000 ... 899 999 | Sequencers/graphs | | | |
| 810 000 ... 810 009 | System error messages | | 50<br>150 (1st NCU) | /PLC/DiagBuffer<br>//<configuration-spec. Designation>/PLC/DiagBuffer |
| 900 001 ... 965 999 | HMI PRO sl Runtime | | | |
| 966 000 ... 999 999 | Reserved | | | |

### SourceIDs 1 ... 10

The SourceIDs 1 ... 10 have the following relationship:

| SourceID | SourceURL |
|---|---|
| 1 | /NCK/Channel#1/Partprogram |
| 2 | /NCK/Channel#2/Partprogram |
| 3 | /NCK/Channel#3/Partprogram |
| 4 | /NCK/Channel#4/Partprogram |
| 5 | /NCK/Channel#5/Partprogram |
| 6 | /NCK/Channel#6/Partprogram |
| 7 | /NCK/Channel#7/Partprogram |
| 8 | /NCK/Channel#8/Partprogram |
| 9 | /NCK/Channel#9/Partprogram |
| 10 | /NCK/Channel#10/Partprogram |

## 12.2.7 Parameter specifications in alarm texts

Alarm texts can contain alarm parameters (accompanying values) that specify the cause of an alarm in more detail. These parameters are usually numerical values that are transmitted from an alarm source when the alarm is signaled along with the other alarm data.

How the parameters are incorporated into the alarm text is specified via place holders (parameter specifications) in the language-specific alarm texts, e.g. "Channel **%1** Axis **%2** in machine data **%3** defined for several channels".

When an alarm occurs, the text is replaced by the corresponding parameters, e.g. "Channel **5** Axis **A3** in machine data **4711** defined for several channels".

### Standard parameter specifications

The following table shows the standard parameter specifications:

Table 12-1 Standard parameter specifications

| Parameter specifier | Description |
|---|---|
| %1 | First parameter from the alarm data of the alarm source. |
| %2 | Second parameter from the alarm data of the alarm source. |
| %3 | Third parameter from the alarm data of the alarm source. |
| %4 | Fourth parameter from the alarm data of the alarm source. |
| %5 | Fifth parameter from the alarm data of the alarm source.<br>Only for NCK alarms:<br>First part (up to the separator) of the fourth parameter from the alarm data of the alarm source. |

| Parameter specifier | Description |
|---|---|
| %7 | Seventh parameter from the alarm data of the alarm source<br>Only for NCK alarms:<br>Third part (between the 2nd and 3rd separator) of the fourth parameter from the alarm data of the alarm source. |
| %8 | Eighth parameter from the alarm data of the alarm source<br>Only for NCK alarms:<br>Fourth part (between the 3rd and 4th separator) of the fourth parameter from the alarm data of the alarm source. |
| %9 | Ninth parameter from the alarm data of the alarm source. |
| %0 | Tenth parameter from the alarm data of the alarm source. |
| %Z | As for %1, in the case of S7-HiGraph alarms the step number of the graph is shown. |
| %K | Specially for PLC alarms:<br>Second digit of the decimal alarm ID: 1**2**3456, corresponds to channel number (0 = channel 10). |
| %A | Specially for PLC alarms:<br>Third and fourth digit of the decimal alarm ID: 12**34**56, corresponds to the axis number. |
| %N | Specially for PLC alarms:<br>Fifth and sixth digit of the decimal alarm ID: 1234**56**, corresponds to the signal number. |

## 12.2.8 Opening error file

The errors that occur during the conversion are written to the "alarmtext_conversion.log" or "oem_text_conversion.log" file.

File loc.: /**user**/sinumerik/hmi/log

### Opening the error file




1. Select the "Start-up" operating area.




2. Press the "System data" softkey.
3. Open the "System CF card" folder.
4. Open the storage directory and select the required file.




5. To open the error file, press the "Open" softkey.

   - OR -

   Press the <INPUT> key.



# 12.3 Configuring an alarm log

## Default

Per default, the alarm log contains all alarms and messages - with their incoming and outgoing time stamps - in chronological order since the last boot. The exceptions are messages from the NC part program (msg command).

Contrary to the "Alarm List" or "Messages" window, all of the alarms or messages that are no longer active when the log is displayed are also displayed (historical alarm events).

## Specify the number of results

The alarm log is organized as a ring buffer: When the maximum size is exceeded, the oldest entries are overwritten by the new alarm events. Each incoming or outgoing time stamp represents a separate event.

## Persistent storage (permanent storage)

If required, the alarm log can be set up as a persistent log that also contains alarm events from before the last time it was switched on. This means that the log is then either time controlled or for each alarm event, is backed-up in an internal binary format in the file system.

Depending on the hardware being used, for the NCU, the alarm log is saved on the CompactFlash card or for the PC/PCU, on the hard disk.

| NOTICE |
| --- |
| **Shorter CF card service life**<br>If the alarm log is a continuously written to the CompactFlash card, then only a limited number of write cycles is possible. Therefore, ensure that the storage is only performed when there is a justifiable need! Undo the setting "at every event" if you no longer require that the alarm log is saved.<br>The CompactFlash card has a service life of over 10 years for an average written data volume of approx. 60 MB/day.<br>The alarm log is not saved in the default configuration. |

## Configuring an alarm log

- Via the operator software in the "Diagnostics" operating area
- Via the "slaesvcconf.xml" configuration file

## 12.3.1 Setting alarm logs from the user interface

### Procedure




1. Select the "Diagnostics" operating area.







2. Press the "Alarm log" and "Settings" softkeys.
3. Enter a number in the "Number of entries" field to change the maximum number of administered incoming and outgoing events.

   The alarm log is organized as a ring buffer: When the maximum size is exceeded, the oldest entries are overwritten by the new alarm events. Each incoming or outgoing time stamp represents a separate event.

   You can specify a value between 0 and 32000. Default value is 500.
4. In the "Write mode file" field, under the following entry, select:
   - "off", the changes are not logged (default setting)
   - "for each event", each alarm event immediately initiates that the alarm log is backed-up (persistent save).
   - "time controlled" if the alarm log is to be backed-up again after a certain time. An additional input field called "Time interval" appears in which you can specify a time in seconds.

   You receive a warning if you only save to a CompactFlash card.

### Restart SINUMERIK Operate

The made settings take effect only after a restart of the operating software. To do this, restart SINUMERIK Operate.

## 12.3.2 Loading alarm log via configuration file

### Creating the configuration file

Adjustments are made in file "slaesvcconf.xml". Copy a sample configuration file "oem_alarmprot_slaesvcconf.xml" from the following directory: /**siemens**/sinumerik/hmi/template/cfg.

Procedure:

1. Save or create the file in the /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg directory.
2. Assign the name "slaesvcconf.xml" to the file.

3. Open the file and enter the number of events to be output in the tag <Records type .../>. The default value is 500. The maximum number depends on the storage medium (hard disk or CompactFlash card).
4. Enter the backup mode in the <DiskCare type="int" value="-1"/> tag. The following values are possible:
   **-1**: There is no backup of the alarm log (default setting).
   **0**: Each alarm event triggers an immediate backup of the alarm log (persistency backup).
   **>0**: Rhythm of the log saving in seconds: When there is a change, the log is time-triggered saved every n > 0 seconds (persistency backup).
5. You adapt the filter for the entry type in the <Filter> tag. Please observe the following:
   - An alarm event is only entered in the log when it satisfies the filter criterion.
   - If several filters are specified in succession, they are linked by a logical OR.
   - Several filters must be combined in series with the keyword AND for an AND operation.

**Note**

Each incoming or outgoing event of an alarm or message requires a separate entry, even when they belong to the same alarm or message.

Acknowledgement events are also contained in the alarm log. You require such entries even when these are presently not recognizable in the alarm log.

## Filter properties

A filter consists of the following three elements <identification> <relation> <value>.

| Entry <identification> | Description |
|---|---|
| AlarmID | Alarm number |
| SourceID | Default values for SourceID and SourceURL can be found in the table in Section Range of alarms (Page 184). |
| SourceURL | |
| <Attribute name> | Arbitrary alarm attribute from the "slaedatabase.xml" file, such as "SEVERITY" or "CLEARINFO". |

| Entry <relation> | Description |
|---|---|
| EQUAL | Equal to |
| NOT | Not equal to |
| LOWER | Less than |
| HIGHER | Greater than |

| Entry <value> | Description |
|---|---|
| Numbers | - |
| Character strings | - |

## Cancel criteria

| Cancel criterion (ClearInfo) | Alarm source (Source) | Description |
|---|---|---|
| 0 | /HMI | Alarms that are canceled by the HMI. |
| 1 | /NCK | Alarms that are canceled by power-on of the NCU. |
| 2 | | Conditions are canceled by a hardware reset of the NCU. |
| 3 | | Conditions are deleted by a "CANCEL" command to the NCU. |
| 4 | | Conditions are canceled by the NCK itself. |
| 5 | | Conditions are canceled by an "NC Start" command on the NCU. |
| 6 | | Conditions are canceled by a reset of the mode group (BAG). |
| 7 | | Conditions are canceled by an "NC Reset" command on the NCU. |
| 8 | /PLC | PLC messages of the FB15 (basic program). |
| 9 | | PLC alarms of the FB15 (basic program). |
| 10 | | Dialog alarms of the HMI that are canceled by the "Recall" key [^]. |
| 11 | | Reserved |
| 12 | | S7-PDiag, S7-Graph, S7-HiGraph or other Alarm_S(Q) alarms of the PLC (SFC17/18) with alarm state "not acknowledged" |
| 13 | | S7-PDiag, S7-Graph, S7-HiGraph or other Alarm_S(Q) alarms of the PLC (SFC17/18) with alarm state "acknowledged". |
| 14 | /NCK | Drive alarms via NCK. |
| 15 | | Part program messages. |
| 16 | | SINAMICS Safety Integrated messages |

## Examples

Log all alarms with a ClearInfo not equal to 15, i.e. no part program messages:

```
<CONFIGURATION>
   <Protocol>
     <Filters>
        <Siemens_Filter_01 type="QString" value="CLEARINFO NOT 15" />
     </Filters>
   </Protocol>
</CONFIGURATION>
```

Log all alarms with the SourceURL "/NCK" or "/HMI":

```
<CONFIGURATION>
   <Protocol>
     <Filters>
        <Filter_01 type="QString" value="SourceURL EQUAL /NCK" />
        <Filter_02 type="QString" value="SourceURL EQUAL /HMI" />
     </Filters>
   </Protocol>
</CONFIGURATION>
```

In the <FilePath> tag, adapt the path and file name for the file in which the alarm log is stored persistently:

```
<CONFIGURATION>
   <Protocol>
    <Filters>
      <FilePath type="QString" value="$(HMI_INSTALL_DIR)user/
      sinumerik/hmi/ log/alarm_log/slaepp_" />
    </Filters>
   </Protocol>
</CONFIGURATION>
```

- Path
  Environment variables can also be incorporated in the path, e.g. for the installation directory: $(HMI_INSTALL_DIR).
- File name
  A 4-digit number in the value range 0000 to 0009 and the file extension "hmi" are automatically added by default to the specified file name during operation. The number is automatically increased when:
  - The file has reached the maximum size (DiskCare = 0), or
  - The specified period has elapsed (DiskCare > 0).
    Older files are deleted.
- Number of logs
  - The number of stored logs can be changed via MaxFileNumbers between 2 and 9999 (default: 10).

## Example: Switching off existing filters

To switch off the existing Siemens filter - without overwriting the filter - proceed as follows:

1. An "empty" filter switches off **all** previously set filters.
   Knowledge of the filter designation is not required.
2. For the effect of the filters, the order in which they are set is important:
   siemens → addon → oem → user

Example of the filtering of alarms 700000 to 700010, 700500 to 700599, and the part program messages (CLEARINFO = 15):

```
<CONFIGURATION>
   <Protocol>
    <Filters>
       <Filter_00 value="" type="QString"/>
       <Filter_01 value="CLEARINFO NOT 15 AND AlarmID LOWER 700000"
       type="QString"/>
       <Filter_02 value="CLEARINFO NOT 15 AND AlarmID HIGHER 700010
       AND AlarmID LOWER 700500" type="QString"/>
```

```
        <Filter_03 value="CLEARINFO NOT 15 AND AlarmID HIGHER 700599"
        type="QString"/>
      </Filters>
    </Protocol>
</CONFIGURATION>
```

### Result

For the changes to the protocol settings to take effect, restart SINUMERIK Operate.

## 12.4 PLC alarms with parameters

### Introduction

For alarms that are triggered via the PLC blocks SFC17 and SFC18, a parameter can be transferred each time the block is called. If the "Octet string" data type is selected, up to 12 bytes can be transferred.

With an appropriate configuration of the operating software, a maximum of 12 bytes can be interpreted, e.g. also as an array of bytes or as a structure with any order of simple data types. In this way, several parameters can be displayed in the alarm text.

A parameter description must be stored for each alarm in the operating software. Two aspects must be taken into account with this parameter description:

- How the parameter has to be interpreted correctly (data type and, if necessary, length)
- How the parameter has to be prepared for display (string or number, decimal, hexadecimal, binary, etc.)

Point 1 is irrespective of the language selected for the display,  point 2 can be different depending on the national language.

The possible parameter descriptions are therefore divided into a language-independent parameter statement and a language-dependent format statement and stored in the following directories:

- Parameter statement: /oem/sinumerik/hmi/cfg/ or /user/sinumerik/hmi/cfg/
- Format statement: /oem/sinumerik/hmi/lng or /user/sinumerik/hmi/lng

The names of the files can be freely selected. The parameter statement is added to the configured data of the relevant alarm in the database in the form of the alarm attribute "HMIPROPARAMDESCR2". Because of the language dependency, the format statement is part of the alarm text.

## 12.4.1 Definition of a parameter of the octet string data type

### Parameter statement

The parameter statement describes how one or more parameters can be defined with the maximum 12-bytes long octet string supplied with the alarm PDU.

### Syntax

The following syntax applies for the parameter statement:

| | | |
|---|---|---|
| <Parameter directive ext> | ::= | [% <Parameter description> ] |
| <Parameter description ext> | ::= | <No. of the associated value><Element type><Offset> |
| <No. of the associated value> | ::= | Always 1 (reserved for future applications) |
| <Offset> | ::= | Data type-dependent offset of the parameter within the associated value:<br>• BOOL data type (B):<br>Bit offset within the associated value<br>• All except for BOOL data type (Y, W, X, I, D, C, R):<br>Byte offset within the associated value |

| Element type | Data type |
|---|---|
| Y | BYTE (8-bit, unsigned) |
| W | WORD (16-bit, unsigned) |
| X | DWORD (32-bit, unsigned) |
| I | INTEGER (16-bit, signed) |
| D | INTEGER (32-bit, signed) |
| B | BOOL (1-bit) |
| C | CHAR (8-bit) |
| R | REAL (32-bit) |

### Examples

| | | |
|---|---|---|
| %1W0%1W2 | → | 1st parameter: 1st associated value, WORD at byte offset 0<br>2nd parameter: 1st associated value, WORD at byte offset 2 |
| %1Y0%1Y1%1W2 | → | 1st parameter: 1st associated value, BYTE at byte offset 0<br>2nd parameter: 1st associated value, BYTE at byte offset 1<br>3rd parameter: 1st associated value, WORD at byte offset 2 |

| | | |
|---|---|---|
| %1B0%1B1%1B2%1B3%1I1 | → | 1st parameter: 1st associated value, BOOL at bit offset 0<br>2nd parameter: 1st associated value, BOOL at bit offset 1<br>3rd parameter: 1st associated value, BOOL at bit offset 2<br>4th parameter: 1st associated value, BOOL at bit offset 3<br>5th parameter: 1st associated value, INTEGER (16-bit, signed) at byte offset 1 |

Example of an alarm attribute file (how to create an alarm attribute file, see Changing alarm attributes (Page 176)):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SlAeAlarmAttributes Version="01.00.00.00">
<Types>
  <Type TypeID="32" TypeName="Condition">
  <Category CatID="1" Version="1.0">
    <CatDescr>Alarms of the SINUMERIK 840D sl.</CatDescr>
    <Attributes>
      <Attribute AttrName="MSGTEXT" AttrID="-1" AttrDataType="10"/>
      <Attribute AttrName="HMIPROPARAMDESCR2" AttrID="5012" AttrDataType="10"/>
    </Attributes>
  </Category>
  </Type>
</Types>
<Sources>
  <Source CatLink="1" SourceID="51" SourceURL="/PLC/PMC">
    <Alarms>
      <!-- Alarm 700000 with three BYTE parameters -->
      <Alarm AlarmID="700000">
        <MSGTEXT>my_oem_plc_alarms_context|700000/PLC/PMC</MSGTEXT>
        <HMIPROPARAMDESCR2>%1Y0%1Y1%1Y2</HMIPROPARAMDESCR2>
      </Alarm>
      <!-- Alarm 700001 with three WORD parameters -->
      <Alarm AlarmID="700001">
        <MSGTEXT>my_oem_plc_alarms_context|700001/PLC/PMC</MSGTEXT>
        <HMIPROPARAMDESCR2>%1W0%1W2%1W4</HMIPROPARAMDESCR2>
      </Alarm>
      <!-- Alarm 700002 with three 16-bit INTEGER parameters -->
```

```
      <Alarm AlarmID="700002">
        <MSGTEXT>my_oem_plc_alarms_context|700002/PLC/PMC</MSGTEXT>
        <HMIPROPARAMDESCR2>%1I0%1I2%1Y4</HMIPROPARAMDESCR2>
      </Alarm>
      <!-- Alarm 700003 with four parameters: BYTE -> WORD -> 32-bit INTEGER -> BYTE -->
      <Alarm AlarmID="700003">
        <MSGTEXT>my_oem_plc_alarms_context|700003/PLC/PMC</MSGTEXT>
        <HMIPROPARAMDESCR2>%1Y0%1W1%1D3%1Y7</HMIPROPARAMDESCR2>
      </Alarm>
    </Alarms>
  </Source>
</Sources>
</SlAeAlarmAttributes>
```

## 12.4.2 Definition of the language-dependent formatting

### Format statement

The format statement is part of the alarm text and is used as a placeholder for a parameters to be displayed.

### Syntax

The following syntax applies for the format statement:

| | | |
|---|---|---|
| <Format directive> | ::= | @ <No. of the parameter description><Format specification> @ |
| %[i]x | ::= | Hexadecimal number with i digits |
| %[i]u | ::= | Decimal number without sign with i digits |
| %[i]d | ::= | Decimal number with sign with i digits (incl. sign) |
| %[i]b | ::= | Binary number with i digits |
| %[i][.y]f | ::= | Fixed-point number: Signed value of the form [-]xxx.yyyy, where i specifies the total number of digits including sign and decimal point and y specifies the number of digits after the decimal point. Missing digits both before and after the decimal point are filled with 0. If there are more places after the decimal point than permitted by y, this is rounded off. |

If the optional digit specification i is not specified, then only the number of digits is used required to completely display the number. The same applies when the number of digits specified by i is not sufficient to completely display the number including sign, i.e. i is then ignored.

The number contained in the format statement does not relate to the position of the associated value in the alarm PDU, but to the position of the parameter description in the parameter statement. Any number of format statements can be present in an alarm text.

### Examples

@1%b@ → Value from the first parameter description to be displayed as a binary number, e.g. "1011011"

@2%8X@ → Value from the second parameter description to be displayed as a hexadecimal number with eight digits, missing digits are filled with 0, e.g. "00AF37FE"

@1%7.2f@ → Value from the first parameter description to be displayed as a fixed point number with seven digits (incl. sign and decimal point) and two digits after the decimal point, missing digits are filled with 0, e.g. "-012.34"

Example of an alarm text file (how you create alarm texts, see Creating in-house alarm texts (Page 168)):

```
<!DOCTYPE TS>
<TS>
<context>
  <name>my_oem_plc_alarms_context</name>
  <message>
    <source>700000/PLC/PMC</source>
    <translation>Text with three BYTE parameters: @1%X@, @2%u@, @3%b@</translation>
  </message>
  <message>
    <source>700001/PLC/PMC</source>
    <translation>Text with three WORD parameters: @1%X@, @2%u@, @3%b@</translation>
  </message>
  <message>
    <source>700002/PLC/PMC</source>
    <translation>Text with three 16-bit INTEGER parameters: @1%X@, @2%u@, @3%b@</translation>
  </message>
  <message>
    <source>700003/PLC/PMC</source>
    <translation>Text with four parameters: @1%X@, @2%u@, @3%r@, @4%b@</translation>
  </message>
</context>
</TS>
```

## 12.5 Deactivating a warning

Generally, alarms/messages are not used without alarm texts. If the alarm text is missing, the warning "No text available" is output.

To deactivate the warning, add the <ControlFlags> area to your "slaesvcadapconf.xml" configuration file.

You can copy this area from the original file. The "slaesvcadapconf.xml" file is in the following directory: /**siemens**/sinumerik/hmi/base

## Setting

In the <MissingTextWarning type="bool" value="FALSE"/> tag, deactivate/activate the warning.

| Entry | Meaning |
|---|---|
| TRUE | The warning is displayed. |
| FALSE | The warning is deactivated. |

## Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CONFIGURATION>
        <ControlFlags>
            <MissingTextWarning type="bool" value="FALSE"/>
        </ControlFlags>
</CONFIGURATION>
```

# Data backup 13

## 13.1 Overview

### Data backup

The following times are recommended for performing a data backup:

- After a start-up
- After changing machine-specific settings
- After replacing a hardware component
- For a software upgrade
- Before the activation of memory-configuring machine data

**Note**

A start-up archive is saved as a file of the type ".arc" (archive).

There are various ways of creating and reloading archives via the SINUMERIK Operate user interface.

- Standard archive: Data can be selected specifically in the data tree and backed up by pressing the "System data" softkey.
- Start-up archive: The "Start-up archive" softkey offers the following selection:
  - Create and read-in a start-up archive
  - Create PLC hardware upgrade archive (only SDBs)
  - Create and read in archive of original status

### Storage locations

Archives can be stored in the following directories:

- CompactFlash card at: System CF-Card/**user**/sinumerik/data/archive or System CF-Card/**oem**/sinumerik/data/archive
- PCU: F:hmisl**user**\sinumerik\data\archive or **oem**\sinumerik\data\archive
- All configured logical drives (USB, network drives)

**Note**

**USB FlashDrive**

USB-flash drives are not suitable as persistent memory media.

### Setting line checksums

To read in a data archive, you need, at a minimum, the authorization that is needed to carry out existing machine data assignments in the archive without any errors. The "Manufacturer" authorization is usually used for this. To ensure that the data archive can also be read in with the "Service" authorization, for example, generate the archive beforehand with line checksums. The machine data assignments that exist in the data archive are thus given valid line checksums and are carried out even if the original authorization is lacking.

**Note**

If you change a line checksum in the data archive with a secured machine data assignment, the line checksum becomes invalid when the archive is read in and alarm 12600 is displayed.

The generation of a line checksum in the archive is activated with the following machine data (default):

| MD11230 $MN_MD_FILE_STYLE | | Structure of the machine data backup files |
|---|---|---|
| Bit 0 = 1 | Line checksum is generated | |
| Bit 1 = 1 | MD numbers are generated | |

## 13.2 Creating a start-up archive

### Overview

A start-up archive can be generated, for example, after the control has been commissioned. To do this, control components can be saved individually or jointly. It is also possible to transfer the backed-up data to other controls so that they can be updated to the same status.

**Note**

When generating the start-up archive, programs that have been moved outside the system (network drive) and any global part program memory that has been set up are not taken into account. Only the programs from the NC memory and the local drive are archived.

### Requirement

You require at least access level 2 (service).

## Data of the control components

| Control components | | Data |
|---|---|---|
| NC data | | Machine data<br>Setting data<br>Option data<br>Global (GUD) and local (LUD) user data<br>Tool and magazine data<br>Protection zone data<br>R parameters<br>Zero offsets<br>Workpieces, part programs and subprograms<br>Standard and user cycles<br>Definitions and macros |
| | With compensation data | • QEC - quadrant error compensation<br>• CEC - operation/angularity compensation<br>• EEC - leadscrew pitch / encoder error compensation<br>It only makes sense to archive machine-specific compensation data if the start-up file is reloaded into the same control. |
| | With compile cycles | Compile cycles (*.elf) are displayed, if compile cycles are available. |
| PLC data (including CP) | | OB (organization blocks)<br>FB (function blocks)<br>SFB (system function blocks)<br>FC (functions)<br>SFC (system functions)<br>DB (data blocks)<br>SDB (system data blocks) |
| Drive data | | Archiving the drive data, either in binary or ASCII format. |
| HMI data, all | | |
| HMI data, selection: | | If data is available, you must set the check mark to activate the components. |

| Control components | | Data |
|---|---|---|
| | Texts | User texts, alarm texts |
| | Templates | Individual templates, workpiece templates |
| | Applications | HMI applications, OEM applications |
| | Engineering data | Engineering data |
| | Configuration | Configurations, incl. display of machine data |
| | Help | Help files |
| | Version data | Version data |
| | Logs | Error logs |
| | Program lists | Program lists |
| | Dictionaries | Dictionaries |
| | Data backups | Files located in the data tree of the "System data" in the "HMI data / data backups" directory. |
| | Programs on local drive/ NC Extend | Programs contained in the user memory area.<br>**Note:** Remember that the user data may be large in volume and any archive file that you create from it will require more memory. |
| Comment | | Entry field for possible comments regarding the start-up archive. |
| Created by | | Entry field for the creator as well as creation date of the start-up archive. |

## Procedure




1. Select the "Start-up" operating area.



2. Press the menu forward key.




3. Press the "Start-up archive" softkey.
   The "Start-up" window opens.




4. Activate "Create a start-up archive" and press the "OK" softkey.
   The "Create Start-up Archive" window opens.
5. Select the desired control components.
6. When required, enter a comment as well as your name and the date of creation.




7. Press the "OK" softkey.
   The "Generate Archive: Select Storage Location" window opens.




8. Position the cursor at the required storage location, press the "Search" softkey and enter the required search term in the search dialog. Then press the "OK" softkey if you wish to search for a certain directory or subdirectory.




**Note**: The place holders "*" (replaces any character string) and "?" (replaces any character) make it easier for you to perform a search.

- OR -




Select the required storage location, press the "New directory" softkey, enter the required name in the "New Directory" window and press the "OK" softkey to create a directory.







9. Press the "OK" softkey.

   The "Generate Archive: Name" window opens.




10. Enter the required name and press the "OK" softkey.

    An archive file with the ARC format type is stored in the selected directory.

## 13.3 Reading-in a start-up archive

| NOTICE |
| --- |
| **Loss of additional languages during software upgrade** |
| All the system languages are overwritten during software upgrades. If there are more languages in the system than in the upgrade archive, these additional languages are lost when the archive is read in. It is not possible to back up the languages before software upgrade. |

**Note**

The reading in of a start-up archive with drive configurations results in the overwriting of the drives in the logdrive.ini file. The system creates a backup copy with the name logdrive_ini.sav or logdrive.oldstyle. Before the archive is read in, the new drive configurations are activated.

**Note**

To prevent the NCU from shutting down unintentionally, you must disable all energy saving profiles before you read in a start-up archive. For more detailed information about the energy saving profiles topic, see:

Operating Manuals for SINUMERIK Operate, Chapter "Ctrl Energy".

### Requirement

You require at least access level 2 (service).

### Procedure




1. Select the "Start-up" operating area.



Press the menu forward key.




2. Press the "Start-up archive" softkey.




3. Activate "Read-in a start-up archive" and press the "OK" softkey.

   The "Select Start-up Archive" window opens.

4. Select a storage location and position the cursor at the required start-up archive (ARC).

   - OR -




   Select the required storage location, press the "Search" softkey and enter the required search term. Then press the "OK" softkey if you wish to search for a certain directory, subdirectory or start-up archive.







5. Press the "OK" softkey.

   The "Read-in Start-up Archive" window opens.

   You obtain an overview with path details, version information, name of the archive, etc., as well as a list of the archived components.

   Remove the tick of a checkbox to exclude a component from the read-in process.

6. Press the "OK" softkey to download the selected archive file.

   The "Read-in Archive" window opens and a progress message box appears for the read-in process.




7. Press the "Overwrite all" softkey if you wish to overwrite the existing files.

   - OR -




   Press the "No overwriting" softkey if you wish to keep existing files.

   - OR -




   Press the "Skip" softkey if you only wish to overwrite certain files.

   You will then obtain a "Read error log for archive" in which the skipped or overwritten files are listed.




8. Press the "Cancel" softkey to cancel the read-in process.

## 13.4 Backing up the hardware configuration

The hardware is configured via the system data blocks (SDB) and only these are stored in the archive.

### Procedure




1. Select the "Start-up" operating area.



Press the menu forward key.




2. Press the "Start-up archive" softkey.
   The "Start-up" window opens.




3. Activate "Create PLC hardware upgrade archive (SDBs only)" and press the "OK" softkey.
   The "Create PLC hardware upgrade archive (SDBs only): Select Storage Location" window opens.
4. Select the desired storage location in the displayed data tree.
   - OR -




   Press the "New directory" softkey to create a separate directory.
   The "New Directory" window opens.




5. Enter the required name and press the "OK" softkey.
   The directory is created below the created folder.




6. Press the "OK" softkey.
   The "Generate Archive: Name" window opens.




   The file type of the archive is ARC and is displayed in the window. Enter the desired name and press the "OK" softkey.
   The archive is generated and stored in the selected directory.

## 13.5 Creating an archive with original data

You can select this type of start-up in order to save the original status of the control. The files are archived in the "original.arc" file.

The file is stored on the CompactFlash card in the Archives/Manufacturer directory.

If there is no "original.arc" file in the directory, it is recommended that an archive be generated from the data of the original control state as delivered.

As for "Create start-up", control components can be saved individually or jointly.

### Requirement

In order to save NC data, you require access level 2 (service).

To back up programs/workpieces, you require access level 6 (keyswitch position 1).

## Data of the control components

| Control components | | Data |
|---|---|---|
| NC data | | Machine data<br>Setting data<br>Option data<br>Global (GUD) and local (LUD) user data<br>Tool and magazine data<br>Protection zone data<br>R parameters<br>Zero offsets<br>Compensation data<br>Workpieces, part programs and subprograms<br>Standard and user cycles<br>Definitions and macros |
| | With compensation data | • QEC - quadrant error compensation<br>• CEC - operation/angularity compensation<br>• EEC - leadscrew pitch / encoder error compensation<br>• It only makes sense to archive machine-specific compensation data if the series start-up file is reloaded into the same control. |
| | With compile cycles | Compile cycles (*.elf) are displayed if compile cycles are available. |
| PLC data (including CP) | | OB (organization blocks)<br>FB (function blocks)<br>SFB (system function blocks)<br>FC (functions)<br>SFC (system functions)<br>DB (data blocks)<br>SDB (system data blocks) |
| HMI data, all | | |
| HMI data, selection: | | |
| | Cycle storage | Definitions and cycles |
| | Texts | User texts, alarm texts |
| | Templates | Individual templates, workpiece templates |
| | Applications | HMI applications, OEM applications |
| | Engineering data | Engineering data |
| | Configuration | Configurations, incl. display machine data |
| | Help | Help files |
| | Version data | Version data |
| | Logs | Error logs |
| | User views | Individually compiled tables with selected machine and setting data. |
| | Dictionaries | Dictionaries |
| | Programs on local drive/ NC Extend | Programs contained in the user memory area.<br>**Note:** Remember that the user data may be large in volume and any archive file that you create from it will require more memory. |

### Procedure




1. Select the "Start-up" operating area.



2. Press the menu forward key.




3. Press the "Start-up archive" softkey.
   The "Start-up" window opens.




4. Activate the "Create archive of original status" checkbox and press the "OK" softkey.
   The "Create Start-up for Original Status" window opens.
5. Select the desired control components.




6. When required, enter a comment into the field as well as your name and the date of creation - and then press the "OK" softkey.
   A warning is output if an "original.arc" archive file already exists.




7. Press the "OK" softkey to overwrite the file.
   A new archive file is generated and stored in the Archives/Manufacturer directory.
   - OR -




   Press the "Cancel" softkey so that the existing file is not overwritten.

## 13.6 Reading in an archive with original data

The data of the original controller state is stored in the "original.arc" file. If you want to restore the controller to the as-delivered state, you can read in the original data archive.

### Precondition

You require access level 2 (service).

### Procedure




1. Select the "Start-up" operating area.



2. Press the menu forward key.




3. Press the "Start-up archive" softkey.
   The "Start-up" window opens.




4. Activate the "Read in archive of original status" checkbox and press the "OK" softkey.
   SINUMERIK Operate automatically accesses the "original.arc" file.
   A window opens prompting whether you want to perform a start-up.




5. Press the "OK" softkey.
   The read-in process is started and a progress message box is displayed in the window.
   Any errors that occur are then displayed in the "Write Error Log for Archive" window.

## 13.7 Generating the complete archive

You have the option of generating an archive of the complete data of your control on an external storage medium that you can provide to Technical Support for diagnostic purposes.

### Procedure

1. Insert the storage medium in the USB port.
2. Press the key combination <Ctrl> + <Alt> + S.
   All data required for diagnostics are compiled in an archive.
   Note:
   Operator control is not possible while the archive is being generated.
3. You will receive a message once archiving has been completed.
   The system assigns an archive name and this is: CompletArchiv<Date>_<Time>.arc.
   The dialog box is closed and you can re-operate the control.

## 13.8 Serial interface (V24 / RS232)

### 13.8.1 Reading-in and reading-out archives via a serial interface

You have the option of reading out and reading in archives in the "Program manager" operating area as well as in the "Startup" operating area via the V24 serial interface.

## Availability of the V24 serial interface

If you want to change the availability of the V24 interface, you can adjust the following parameters in file "slpmconfig.ini":

| Parameters | Description | |
|---|---|---|
| [V24] | Describes the section in which the relevant setting parameters are located. | |
| useV24 | Setting for the availability of the V24 serial interface | |
| | = true | Interface and softkeys are available (default) |
| | = false | Interface and softkeys are not available |

## Storage of file "slpmconfig.ini"

The template of the file "slpmconfig.ini" for SINUMERIK Operate is stored in the following directory:

<Installation path>/siemens/sinumerik/hmi/template/cfg

Copy the file to one of the following directories:

<Installation path>/**user**/sinumerik/hmi/cfg

<Installation path>/**oem**/sinumerik/hmi/cfg

**Note**

If you want to achieve a better overview of the changes you have made yourself, simply delete the unchanged parameters from the file copy "slpmconfig.ini".

## Reading-out archives

The files to be sent (directories, individual files) are zipped in an archive (*.arc). If you send an archive (*.arc), this is sent directly without being additionally zipped. If you have selected an archive (*.arc) together with an additional file (e.g. directory), then these are zipped into a new archive and are then sent.

## Reading-in archives

Use interface V24 if you want to read in archives. They are transferred and then subsequently unzipped.

**Note**

**Reading in commissioning archives**

When you read in a commissioning archive via the V24 interface, then this is immediately activated.

## Externally processing the punched tape format

If you wish to externally process an archive, then generate this in the punched tape format.

## Procedure




1. Select the "Program manager" operating area, and press the "NC" or "Local. drive" softkey.




...




- OR -




Select the "Startup" operating area and press the "System data" softkey.




### Reading-out archives

2. Select the directories or the files that you wish to send to V24.



3. Press the ">>" and "Archive" softkeys.







4. Press the "V24 send" softkey.

- OR -

### Reading in an archive




Press the "V24 receive" softkey if you wish to read-in files via V24.

## 13.8.2 Setting interface parameters

### Procedure




1. Select the "Start-up" operating area.




2. Press the "System data" softkey.






3. Press the ">>" and "Archive" softkeys.




4. Select the "RS-232-C settings" softkey.
   The "Interface: RS-232-C" window is opened and the interface settings are displayed.




5. Press the "Details" softkey if you wish to view and process additional settings for the interface.

### RS-232-C settings

| Parameter | Meaning |
|---|---|
| Protocol | The following protocol is supported for transfer via the RS-232-C:<br>• RTS/CTS |
| Transfer | It is also possible to use a secure protocol for data transfer (ZMODEM protocol).<br>• Normal (default setting)<br>• Secure<br>For the selected interface, secure data transfer is set in conjunction with handshake RTS/CTS. |
| Baud rate | Transfer rate: The baud rate that can be used depends on the connected device, the cable length and the general electrical conditions.<br>• 19200 (default setting, maximum baud rate for NCU (Linux))<br>• 115200 (maximum baud rate for PC/PCU (Windows)) |
| Archive format | • Punched tape<br>• Binary format (PC format) |
| **RS-232-C settings (details)** | |
| Interface | • COM1<br>• COM2 (only relevant for SINUMERIK Operate on PC) |

| Parameter | Meaning |
|---|---|
| Parity | Parity bits are used for error detection: The parity bits are added to the coded characters to make the number of positions set to "1" an uneven number (uneven parity) or to an even number (even parity).<br>• None (default setting)<br>• Odd<br>• Even |
| Stop bits | Number of stop bits for asynchronous data transfer.<br>• 1 (default setting) |
| Data bits | Number of data bits for asynchronous data transfer.<br>• 8 bits (default setting) |
| XON (hex) | Only for punched tape format |
| XOFF (hex) | Only for punched tape format |
| End of data transfer (hex) | Only for punched tape format<br>Stop with end of data transfer character<br>The default setting for the end of data transfer character is (HEX) 1A. |
| Time monitoring (sec) | Time monitoring<br>For data transfer problems or at the end of data transfer (without end of data transfer character) data transfer is interrupted after the specified number of seconds.<br>The time monitoring is controlled by a time generator (clock) that is started with the first character and is reset with each transferred character. The time monitoring can be set (seconds). |

## 13.9 Backing up setup data

### Setting for "Backup setup data"

The "Backup setup data" function should only be used if MD11280 $MN_WPD_INI_MODE is set to 1 (default setting). Otherwise, when the NC is started for the first time after selecting the part program, a file with the same name with the ".ini" extension would be automatically executed.

Machine data:

| MD11280 $MN_WPD_INI_MODE | | |
|---|---|---|
| Processing mode of ini files in the workpiece directory | | |
| = 1 | When the NC is started for the first time, after the workpiece selection, ini files with the name of the selected part program and the following extensions are executed: | |
| | CEC | Sag compensation (Cross Error Compensation) |
| | GUD | User data (Global User Data) |
| | PRO | Protection zones |
| | RPA | R parameters |
| | SEA | Value assignments (Setting Data Active) |
| | TMA | Magazine data (Tool Magazine Active) |
| | TOA | Tool offsets (Tool Offset Active) |
| | UFR | Work offsets (User Frame) |

## 13.10 Network settings

### Activating

By default, the network settings cannot be backed up. The archiving of the network settings must be enabled:

1. Copy the "slpmconfig.ini" file from the /siemens/sinumerik/hmi/template/cfg directory.
2. Store the file in the /oem/sinumerik/hmi/cfg or /user/sinumerik/hmi/cfg directory.
3. Open the file and enter the following for [SeriesSetup]:
   SupportNetworkSettings=true
4. Restart SINUMERIK Operate.

**Note**

This setting also affects the creation of a complete standard archive using Ctrl+Alt+S or Ctrl +Alt+C.

### References

For descriptions of key combinations, refer to the following documentation:

Operating Manual Universal/Turning/Milling/Grinding, chapter "Introduction" > "Operator panel front"

"Keys of the operator panel"

## Restrictions

- The leases files are not saved:
  - /system/etc/udhcpd-eth0.leases
  - /system/etc/udhcpd-ibn0.leases
- A configuration in the system network must be based on the DNS name rather than IP addresses.

## Basic procedure

- Backup of network settings
  After activation, the network settings can also be backed up when a commissioning archive is created when a check mark is set, see Creating a start-up archive (Page 200).
- Importing network settings
  The saved archives can be imported in the same manner as a commissioning archive, see Reading-in a start-up archive (Page 203).

**Note**

**Importing network settings for replacement components**

Disconnect the replacement components from the system network before you import the saved network settings.

## NCU

With SINUMERIK Operate on the NCU, the following NCU settings are backed up:

- /user/system/etc/basesys.ini
- /user/common/tcu (complete directory)

**Note**

The network settings of an NCU can only be backed up and read on the PG/PC via SINUMERIK Operate.

The NC address in mmc.ini on the PC/PG is 192.168.215.1.

## PCU

The following PCU settings and, if applicable, NCU settings are saved with SINUMERIK Operate on the PCU:

- PCU
  - /user/system/etc/basesys.ini
  - mmc.ini for SINUMERIK Operate
- NCU (when present as master)
  - /user/system/etc/basesys.ini
  - /user/common/tcu (complete directory)

The following archives are created:

- <Archive_name>_pcu.arc
- <Archive_name>_ncu.arc

**Importing an archive for a configuration with NCU as master (DHCP) and PCU (on_low):**

1. Import the archive for Archivename_pcu.arc on a PCU with SINUMERIK Operate.
2. Depending on the replaced parts of the NCU, note the following:
   - If the NCU was not replaced, the system restarts.
   - If the NCU has been replaced, <Archive_name>_ncu.arc must be read in via a separate user interface with the X127 interface.

# Configuring the network

# 14

## Overview

If you wish to perform network settings in SINUMERIK Operate, you have the following options:

- The Start-up operating area allows you to set up all of the basic configurations for the plant and company networks (see Section "Displaying the network overview (Page 217)").
- The Diagnostics operating area gives you an effective overview of all available network adapters and allows you to assign and check settings for the company network or the DHCP server, for example (see Section "Configuring TCP/IP diagnostics (Page 238)").

## 14.1 Displaying the network overview

All the available interfaces and their network settings are listed in the "Overview of the network settings" window. For example, the IP addresses, subnet masks and MAC addresses for the company and system network are displayed here. You have the option of editing the settings.

## Procedure




1. Select the "Start-up" operating area.



2. Press the "Menu forward" key.




3. Press the "Network" softkey.




The "Overview of the Network Settings" window opens.




4. Press the "Change" softkey to edit settings.

## See also

Settings of the system network (Page 218)

Settings of the factory network (Page 221)

Saving network settings (Page 223)

## 14.2 Settings of the system network

The configuration for the DHCP server, the routing, active protocols, for DNS servers and other properties of the system network is performed in the "Settings of the system network" window.

| Setting | Description |
|---|---|
| DHCP activated | Activates or deactivates the DHCP mode for the system network. This automatically assigns IP addresses to the nodes in the system network. |
| DHCP server synchronization mode | Activates or deactivates the DHCP server synchronization mode. If this setting is active, then the DHCP servers synchronize themselves in the system network (X120) so that only one of them actively assigns addresses. This makes it possible to operate multiple NCUs or PCUs concurrently without having to adjust the network settings.<br>The DHCP servers that are not active go into "standby" mode; in this mode they regularly get the current address data and TCU data from the active server so that if the active server fails, a standby server can take over the active role without data loss.<br>• With the "Master priority" setting you can influence the synchronization in such a manner that the server with the "Master" setting is always the active server (it must be active in the network). This means that in the normal situation, deterministically the same control is always the DHCP server and you can find the actual address data and the TCU data there. "Master priority" must only be set for a single DHCP server in the system network.<br>• The settings "Low priority " and "High priority" can be used to specify which DHCP server takes precedence as the active server. A DHCP server with low priority is then only used as active server when a server with high priority cannot be found in the network.<br>Default: "High priority" |
| DHCP address range | Specifies the range for DHCP addresses. The range starts at the address specified at "Start" and ends at the address entered at "End". The resulting range of numbers has the following default setting:<br>Start address = first address in the system network<br>End address = start address + 10 or + 2 (if less than 16 addresses are available) |

| Setting | Description |
|---|---|
| Timeout when waiting for master | Specifies a maximum waiting time (in seconds) for the response of the master server (see DHCP server synchronization mode). After this time elapses, the server itself becomes the active server.<br>This additional pause makes it possible for the intended DHCP master to become the active DHCP server without displacement even if it is shortly switched on thereafter, or takes longer to boot than other controls.<br>Default: 120 seconds |
| Host name | Assigns the DHCP server a fixed host name. In this way, the server can also be addressed via the host name and not only via the IP address.<br>Because the host name is also used as DNS name, it must satisfy the corresponding requirements of the RFC:<br>• ASCII letters (a-z, A-Z), digits (0-9) and '-'<br>• maximum 63 characters |
| DNS domain | This is used to specify the Top Level Domain (TLD) name used in the system network. The DNS server of the NCU assigns names to the devices in the system network in this zone. Name requests for all other zones are forwarded to an external name server in the company network.<br>The "local" default setting recommended by the RFC1035 is used for local networks, to avoid conflicts with globally defined domain names. As a rule, this default setting is sufficient. |
| PN station name | A separate machine name can be defined here for PROFINET purposes. This is used at the ERTEC interface (when available), transferred to the ERTEC system program and also used by DCP at this network interface. |
| Station name | This string is used for the SINUMERIK-specific OID mcSinumerikMIB.mcSinumerikMiscStation. The value is the name of a station to which the device belongs. Devices with the same station name can therefore be identified as belonging together. This is for information purposes only. |
| Location | This string is used for the standard OID SNMPv2-MIB::sysLocation. If required, a location can be specified here which can be called by an SNMP client. This is for information purposes only. |
| Contact person | This string is used for the standard OID SNMPv2-MIB::sysContact. If required, a contact address can be specified here which can be called by an SNMP client. This is for information purposes only. |

| Setting | Description |
|---|---|
| Function | This string is used for the Siemens Industry-specific OID automationSystemIdent.automationFunctionTag. If required, a function designation can be specified here which can be called by an SNMP client. This is for information purposes only. |
| Fixed domain | Defines an additional DNS domain name (suffix for DNS queries). This is of especial interest when the DHCP server is switched off at the X120 because this cannot set the "local" domain, but it is required for some DNS queries (e.g. for machine control panels). |
| Fixed DNS server | Specifies up to three DNS servers that are to be used by the DHCP server. This is of especial interest when the DHCP server is switched off at the X120 because this is also a DNS server or determines DNS servers in the network. |
| Routing | |
| • Routing X120/X127 -> X130 | Specifies whether the routing of packets from the system network (X120) and/or service connection X127 to the company network (X130) is active:<br>• Released: Activates the routing for X120 and X127<br>• Disabled: Deactivates the routing for X120 and X127<br>• X127: Activates only the routing from X127 to the company network<br>• X120: Activates only the routing from the system network to the company network |
| • Routing X127 -> X120 | Activates or deactivates the NAT routing of packets from the service connection X127 to the system network (X120). |
| • Routing X120 -> X127 | Activates or deactivates the routing of packets from the system network (X120) to the service connection X127. This is normally prohibited by the firewall.<br>Note that there is no NAT and the sender must ensure that packets to the PG or service PC at X127 with address 192.168.215.x actually reach the NCU to which the device is connected. |
| Active protocols | Specifies which protocols are to be used in the system network. "DCP" (Discovery and Basic Configuration Protocol) and "LLDP" (Link Layer Discovery Protocol) are available for selection. The appropriate active protocol is also used for the service connection X127 at the NCU. |

### Procedure




1. The network overview is open.




2. Press the "System network" softkey.
   The "Settings of the System Network" window opens.




3. Press the "Change" softkey to edit settings.




4. Press the "Reset DHCP server" softkey to return the current settings of the DHCP server to the original settings.




5. Press the "Distribute DHCP data" softkey to synchronize the settings with all nodes in the system network.

**Note**

You can only use the "Reset DHCP server" and "Distribute DHCP data" softkeys with a connection to the active DHCP server.

## 14.3 Settings of the factory network

The network interface, active protocols and firewall exceptions of the company network are configured in the "Settings of the company network" window.

| Setting | Description |
|---|---|
| Gateway | If this value is not empty, the host specified there is used as the default gateway, which means that all IP packets that cannot be directly assigned are sent there for routing. |
| DNS server | If DNS name servers are specified here (maximum three), then they must be used to resolve symbolic host names, i.e. at most of the points where an IP address is expected, a computer name can also be used instead.<br>The name server setting is also sent to its DHCP clients (TCU, PG) via the DHCP server of the NCU, so that they can also work with symbolic names. |
| Time server | Here you can specify up to three NTP servers (UDP/123) that can be used by the NTPD on the NCU for time synchronization. |

| Setting | Description |
|---|---|
| Host name | Here you can define a name for the local host. This manually assigned name takes priority over all other names.<br>The host name is determined in the following order:<br>• This entry (identical to the host name from the basesys.ini file)<br>• A name received from the DHCP server (reverse DNS lookup, i.e. which name belongs to the received IP address)<br>• Default name ("NONAME_...")<br>Because the host name is also used as DNS name, it must satisfy the corresponding requirements of the RFC:<br>• ASCII letters (a-z, A-Z), digits (0-9) and '-'<br>• maximum 63 characters |
| Active protocols | Specifies which protocols are to be used in the company network. DCP (Discovery and Basic Configuration Protocol) and LLDP (Link Layer Discovery Protocol) are available for selection. |
| Firewall exceptions | You can specify ports here that are to be enabled in the external network interface. |
| • S7 communication (TCP/102) | Port for the S7 communication |
| • VNC access (TCP/5900) | Port for the VNC access |
| • Additional ports | Any further ports can be enabled here. The entry starts with the protocol name (TCP or UDP), followed by a slash '/' and the port number of the service to be enabled. Several entries are separated by a space.<br>Example (enabling of the VNC servers): `TCP/5900 TCP/5904 TCP/5905` |

**NOTICE**

**Security risk**

Note that opening the ports in the firewall can represent a security risk. Only enable the ports that you actually require.

Note that every accessible service can have security gaps.

### Procedure




1. The network overview is open.




2. Press the "Company network" softkey.
   The "Settings of the Company Network" window opens.




3. Press the "Change" softkey to edit settings.

## 14.4 Saving network settings

You have the option of saving the NCU settings as file.

In window "Save network settings: Select archive", select a target folder. The network configuration is saved there, or an existing file is selected, which is overwritten with the new settings.

### Procedure




1. The network overview is open and you have made network settings.




2. Press the "Save NCU settings" softkey.
   Window "Save network settings: Select archive" is opened.
3. Select the folder in which the network configuration is be stored. If a file is already available, it will be overwritten.
   - OR -
   Press the "New directory" softkey to create a new directory.
   - OR -




   Press the "Find" softkey, for example, to find the archive location of the network configuration file.




4. Press the "Find next" softkey if the file found during the search does not correspond to the required configuration file.




5. Press the "Cancel" softkey to reject the settings.

   - OR -




   Press the "OK" softkey to save the network configuration.

# 14.5 Configuring operator panels

## 14.5.1 Displaying the available operator panels

A list with all the available operator panels is displayed in the upper part of the **Available Operator Panels ...** window. One line corresponds to one operator panel and contains the following information:

| Column | Description |
|---|---|
| | Consecutive number to identify the operator panel |
| Status | Status of the operator panel:<br> : The operator panel is available on the bus.<br>: The operator panel is not available on the bus. |
| HW | Information on the hardware of the operator panel<br>Possible values are unknown (no information available), NCU, PCU, PPU, TCU (all TCU including HT 8, etc.).<br>Default value: Unknown |
| Name | Name of the operator panel |
| MCP | DIP switch position for the machine control panel<br>The values can be freely selected from 1 to 254 (0 means that the last active machine control panel is retained, 255 means that a machine control panel is not being used).<br>Default value: 192 |
| DCK | Availability of direct control keys<br>Active: The TCU has direct control keys<br>Inactive: The TCU has no direct control keys |
| TCU | Index for the TCU (Thin Client Unit)<br>The value is freely configurable from 0 to 255 and is used for identification in the PLC. |
| EKS | DIP switch position for the chip reader for access levels (Electronic Key System)<br>Possible values range from 1 to 255. The value 0 means that an EKS is not used. |
| Virtual key-board | Use of a virtual keyboard<br>Possible values are "Never", "Always" and "Auto", whereby "Auto" is set as default value. |

**Note**

The information of the operator panels is read-only and cannot be changed in the **OPs** window.

### Update of the overview

When switching to the **OPs** window, a search for available operator controls is performed automatically. This is also possible later via the "New display" or "Reload OP data" softkeys. The latter action results in a completely new scan during which already available operator controls are deleted from the local cache. New operator controls and data are sought during a reload.

The active DHCP server that contains the data for the operator panels can be a different NCU, but also the local NCU itself. Irrespective of this, a copy of the data is stored and used on the local NCU.

### Procedure




1. Select the "Start-up" operating area.




2. Press the "OPs" softkey.

   The "OPs" window is opened.

   All the presently available operating elements are displayed.







3. Press the "New display" softkey to refresh the overview.

   A search is performed for new operator controls and data.




4. Press the "Reload OP data" softkey to start a completely new search.

   Already available operator controls are deleted from the buffer during this search.




5. Press the "Distribute TCU data" softkey to synchronize the available controls with all participants in the system network.

**Note**

You can only use the "Distribute TCU data" softkey with a connection to the active DHCP server.

## 14.5.2 Configuring VNC connections

A list of the available VNC connections is displayed in the lower part of the "Available Operator Panels ..." window. It contains the following information:

| Column | Description |
|---|---|
| | Consecutive number to identify the VNC connection |
| VNC server | IP address or name of the NC |
| Displacement | Specifies whether it is possible that the TCU will be displaced by other nodes. |
| /Priority | Displacement priority: Specifies the priority which influences whether the TCU will be displaced by other nodes.<br>The possible values lie between 0 and 10. The smaller the value, the faster the TCU will be displaced. If the displacement is blocked, the priority will be ignored. |
| Start | Specifies whether the TCU is to be started active, i.e. as operator panel. |
| /Priority | Start-up priority: Specifies the priority of the TCU connection to be used.<br>Any value greater than 0 is permitted. The larger the value, the lower the connection priority. |

| Column | Description |
|---|---|
| Display area | Specifies which HMI screen is displayed when activating the TCU. |
| Channel | Specifies which channel is displayed when activating the TCU. |

### Editing the list of VNC connections




1. The "Available Operator Panels ..." window is open.




2. Press the "Change" softkey.
   The information on the VNC connections can be edited.

   - OR -




3. Press the "Add" softkey to establish a new VNC connection.

   - OR -




4. Press the "Remove" softkey to remove a selected VNC connection from the list.

   - OR -







5. Press the "Move up" or "Move down" softkey to move a VNC connection within the list.




6. Make the required change and press the "OK" softkey.

   - OR -




   Press the "Cancel" softkey to cancel the changes.

### See also

Displaying the available operator panels (Page 224)

## 14.5.3 T:M:N configuration

By T:M:N configuration we mean a grouping of t operating stations (TCUs with MCP), m HMI systems and n NCK systems, in which the operating stations can be freely switched to the various HMI systems in order to operate. In a two-stage process, the switchover targets for each operating station are individually configured in a channel menu.

From the point of view of the HMI, T:M:N operation is active if a channel menu is configured. Available switchover targets are defined directly via the address of the HMI and optionally via an NC channel, which must be assumed on switchover from the HMI. These switchover targets can be combined into so-called channel groups. Channel groups are displayed on horizontal

softkeys in the channel menu, the switchover targets themselves (HMI, NC channel) are displayed on the vertical softkeys assigned to a channel group.

### Linking of softkeys to channel groups

A list for the linking of the HMI softkeys to configured channel groups is displayed in the lower part of the window. The channel groups can in turn contain switchover targets that execute functions of the operator panel, for example. Up to 32 softkeys (SK_1 to SK_32) can be linked to a specific channel group (e.g. CH_GRP_0).

### Procedure




1. The "Available Operator Panels ..." window is open.




2. Select the required operator panel and press the "T:M:N" softkey
   A list for the linking of the HMI softkeys to configured channel groups is displayed in the lower part of the window.




3. Press the "Change" softkey.
   The list can be edited.




4. Make the required change and press the "OK" softkey.

   - OR -




   Press the "Cancel" softkey to cancel the changes.

### See also

Displaying the available operator panels (Page 224)

## 14.6 Station-related network diagnostics

Using the station-related network diagnostics, you have the possibility of identifying faulty or incorrectly parameterized components or those components that have not been connected.

You obtain an overview of all of the components belonging to the system network:

- NCU
- PCU
- TCU
- MCP
- EKS

If components are either faulty or absent (or unavailable), they will be marked as such in this overview.

#### Error analysis

You have the option of initiating a fault analysis for the components involved. You obtain the result of a such a diagnostics in the form of a fault message. The possible fault states and causes are described in this fault message together with how to resolve the fault.

## 14.6.1 Displaying network adapters

The actual network adapters (system network X120, company network X130 or system network ETH2 and company network ETH1) and their availability are displayed in a tree-like structure in the "TCP/IP diagnostics" window.

#### Displaying the components

The following components are displayed in the window:

- The control is displayed at the topmost level.
- System network (X120 / ETH2) with configured operator stations with IP address
  - Operator panels with IP address
  - Machine control panels
  - EKS
- Company network (X130 / ETH1) with
  - Operator panels with IP address
  - Machine control panels
  - EKS

#### Components that cannot be accessed



If a component cannot be accessed, then it is marked using this symbol.

#### Input rights

Operator panels, which have input rights, have a green background.

#### Detailed view

You have the option to display the following information about a selected component.

#### TCU

- IP address
- SW version
- MCP index from the configuration
- TCU index from the configuration
- DNS name
- Resolution

**MCP**

- IP address
- DNS name
- MCP index from the configuration
- MCP index requested from HMI
- MCP index requested from PLC

## Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key and the "Bus TCP/IP" softkey.







3. Press the "TCP/IP diagnostics" softkey.
   The "TCP/IP Diagnostics" window opens and displays the current availability of the network connections.
4. Position the cursor on the component for which you wish to display the detailed information.




5. Press the "Details" softkey to display all available parameters of the configured network connections.




6. Press the "Details" softkey to hide the detailed display.




7. Press the "Return" softkey in order to exit the "TCP/IP diagnostics" window and return to the "PROFIBUS diagnostics" window.

## 14.6.2 Error analysis

### 14.6.2.1 Performing error diagnostics

## Precondition

Error analysis only functions for a PCU.

### Procedure



1. The "TCP/IP Diagnostics" window opens.
2. Position the cursor on the component that is marked as having an error (NCU).




3. Press the "Error analysis" softkey.
   The corresponding error message is displayed.

## 14.6.2.2 MCP/EKS as network participant cannot be accessed

| Error | Possible error causes and remedies |
|---|---|
| The MCP/EKS cannot be accessed as network participant. | |
| | **Error cause 1** |
| | There is no physical network connection.<br>**Notice**: The display only refers to the circumstance that the direct connection from MCP/EKS to the next connection nodes has not been established, i.e.<br>• The Ethernet cable is not directly inserted at the MCP/EKS,<br>• The Ethernet cable is not inserted at the direct connection partner or the partner is not available (e.g. MCP),<br>• The Ethernet cable connected to the MCP/EKS is defective. |
| | **Remedy** |
| | Establish the physical network connection (check the plug-in connections, replace the cable, check the connection partner). |
| | **Error cause 2** |
| | The physical network connection does not exist after the next connection node to the system network. There is no physical connection from the MCP/ECS to the computer with the active DHCP server. The cables are either defective, have not been inserted or switches have not been closed. Switches can also be located on the MCP or distribution box. |
| | **Remedy** |
| | Establish the physical network connection (check the plug-in connections, replace the cable, check the connection partner).<br>Once the connection has been established, the TCU continues to boot. |
| The MCP/EKS cannot be accessed as network participant. There is a physical network connection to the system network. | |
| | **Cause of the error** |
| | No active DHCP server is found in the system network. Possible causes:<br>• Only a single computer has been configured in the system network with active DHCP server - and this computer is not available. The system network is not operated with synchronizing DHCP servers.<br>• No computer has been configured in the system network with active DHCP server. DHCP operation is disabled on all potential computers (NCU, PCU). This is a configuration error.<br>• In the system network, there is no NCU or PCU computer available that has DHCP operation activated on it and could be a server. |
| | **Remedy** |
| | In the system network, configure a computer that is available as DHCP server. |

### 14.6.2.3 HMI on the PCU cannot establish a network connection to the NC

| Error | Error cause 1 | Remedy |
|---|---|---|
| HMI displays alarm 120202 "Wait for connection to NC/ PLC" and/or no values (only '#'). | The IP address of the target NCK/PLC set on the PCU is incorrect. | From the HMI, correct the target address for NCK/PLC and restart the PCU. |

| Error | Error cause 2.1 | Remedy |
|---|---|---|
| HMI displays alarm 120202 "Wait for connection to NC/ PLC" and/or no values (only '#'). | The PCU has an IP address that does not belong to the network or has an invalid IP address.<br>The TCU support is deactivated on the PCU (= DHCP is disabled) and the PCU either operates with a permanently set IP address that does not belong to the network, or with the standard Windows DHCP client to dynamically retrieve addresses. This is a configuration error. | Correctly set the network settings for the system network in the Windows service mode (set the fixed specified IP address and subnet mask for the system network, use the addressing mode, fixed IP address).<br>Execute the following actions using the "System Network Center" program:<br>• Connect the DHCP service to the interface for the system network<br>• Set the DHCP mode (TCU support, DHCP SyncMode, address range of the addresses that can be assigned)<br>• Initiate DHCP clear |

| Error | Error cause 2.2 | Remedy |
|---|---|---|
| HMI displays alarm 120202 "Wait for connection to NC/ PLC" and/or no values (only '#'). | The PCU has an IP address that does not belong to the network or has an invalid IP address.<br>The TCU support is activated on the PCU (= DHCP is enabled) and the PCU is active DHCP-/DNS server (either as a result of the DHCP SyncMode ON_MASTER or as a result of the Sync priority - just the same as the other server candidates and random selection) - or the PCU is a DHCP client. As a result of a software error or the actual network configuration, the PCU cannot obtain a valid IP address (e.g. '0 address'). | Correctly set the network settings for the system network in the Windows service mode (set the fixed specified IP address and subnet mask for the system network, use the addressing mode, fixed IP address).<br>Execute the following actions using the "System Network Center" program:<br>• Connect the DHCP service to the interface for the system network<br>• Set the DHCP mode (TCU support, DHCP SyncMode, address range of the addresses that can be assigned)<br>• Initiate DHCP clear<br>In addition, at the station with DHCP master (DHCP SyncMode = ON_MASTER) a DHCP clear should be initiated. |

| Error | Error cause 2.3 | Remedy |
|---|---|---|
| HMI displays alarm 120202 "Wait for connection to NC/ PLC" and/or no values (only '#'). | The PCU has an IP address that does not belong to the network or has an invalid IP address.<br>The TCU support is activated on the PCU (= DHCP is enabled) and the PCU is active DHCP-/DNS server (either as a result of the DHCP SyncMode ON_MASTER or as a result of the Sync priority - just the same as the other server candidates and random selection). The PCU is set with an incorrect IP address and an incorrect IP range for the IP addresses that can be dynamically allocated. The PCU possibly changed from DHCP standby to active server as the computer with DHCP MASTER has failed or been switched-off. | Correctly set the network settings for the system network in the Windows service mode (set the fixed specified IP address and subnet mask for the system network, use the addressing mode, fixed IP address).<br>Execute the following actions using the "System Network Center" program:<br>• Connect the DHCP service to the interface for the system network<br>• Set the DHCP mode (TCU support, DHCP SyncMode, address range of the addresses that can be assigned)<br>• Initiate DHCP clear<br>In addition, at the station with DHCP master (DHCP SyncMode = ON_MASTER) a DHCP clear should be initiated. |

| Error | Error cause 2.4 | Remedy |
|---|---|---|
| HMI displays alarm 120202 "Wait for connection to NC/ PLC" and/or no values (only '#'). | The PCU has an IP address that does not belong to the network or has an invalid IP address.<br>TCU boot support is activated on the PCU (= DHCP is enabled) and the PCU is the standby DHCP/DNS server (i.e. in the role of the DHCP client). An additional active DHCP server was switched into the system network while operational; this active DNCP server also has an incorrect address range. This is a configuration error. | Correctly set the network settings for the system network in the Windows service mode (set the fixed specified IP address and subnet mask for the system network, use the addressing mode, fixed IP address).<br>Execute the following actions using the "System Network Center" program:<br>• Connect the DHCP service to the interface for the system network<br>• Set the DHCP mode (TCU support, DHCP SyncMode, address range of the addresses that can be assigned)<br>• Initiate DHCP clear<br>In addition, at the station with DHCP master (DHCP SyncMode = ON_MASTER) a DHCP clear should be initiated.<br>Switch the complete system off and on again. |

| Error | Possible error causes and remedies |
|---|---|
| HMI displays alarm 120202 "Wait for connection to NC/PLC" and/or no values (only '#'). | |
| | **Error cause 3.1** |
| | The NCU has an IP address that does not match the network or has an invalid IP address.<br>The NCU does not participate in DHCP operation and runs with a permanently set IP address that is incorrect and does not match the system network. |
| | **Remedy 3.1** |

| Error | Possible error causes and remedies |
|---|---|
| | Connect the service PG to the NCU via X127. Log into Linux via WinSCP or Putty. Using the service command 'sc show ip -eth0', display the actual settings - and check these. If the cause of the error is confirmed, then the NCU must be reparameterized.<br>Perform the service commands and settings:<br>• sc enable DHCPSvr –X120 or the equivalent sc enable DHCPSvr –eth0 to activate DHCP operation<br>• sc enable DHCPSync –X120 -LOW\|HIGH\|MASTER or equivalent sc enable DHCPSync –eth0 -LOW\|HIGH\|MASTER to set the DHCP synchronization mode of this NCU<br>Enter the following entries correctly in the /user/system/etc/basesys.ini file<br>:[InternalInterface]<br>; With InternalIP and InternalNetMask (both must be set together),<br>; you can change the address on the internal/TCU/automation/system net.<br>InternalIP=192.168.214.1<br>InternalNetMask=255.255.255.0<br>…<br>; This is the range of dynamic IPs given out by the DHCP server<br>; Defaults should be sensible<br>InternalDynRangeStart=192.168.214.10<br>InternalDynRangeEnd=192.168.214.239<br>The values in green should be set corresponding to the overall system.<br>Then restart the NCU. |
| | **Error cause 3.2** |
| | The NCU has an IP address that does not match the network or has an invalid IP address.<br>The NCU participates in DHCP operation and the NCU is an active DHCP/DNS server (either as a result of the DHCP SyncMode ON_MASTER or as a result of the Sync priority - just the same as the other server candidates and random selection). The NCU is set with an incorrect IP address and an incorrect IP range for the IP addresses that can be dynamically allocated. The NCU possibly changed from DHCP standby to active server as the computer with DHCP MASTER has failed/has been switched-off. |
| | **Remedy 3.2**<br>The same as for remedy 3.1 |
| | **Error cause 3.3** |
| | The NCU has an IP address that does not match the network or has an invalid IP address.<br>The NCU participates in DHCP operation and operates with the incorrect IP address. After the NCU booted, an additional component that is already operational (NCU or PCU) with active DHCP server was connected to the network. An additional active DHCP server was switched into the system network while operational; this active DNCP server has an incorrect address range. This is a configuration error. |
| | **Remedy 3.3**<br>The same as for remedy 3.1 |
| HMI displays alarm 120202 "Wait for connection to NC/PLC" and/or no values (only '#'). | |
| | **Error cause 4.1** |
| | The NCU has a valid IP address in the system network, however, not the required IP address.<br>The NCU does not participate in DHCP operation and operates with a permanently set target address that is different from that saved in MMC.INI in the system network - however, this is a valid address. This is a configuration error. |
| | **Remedy 4.1** |

| Error | Possible error causes and remedies |
|---|---|
| | Connect the service PG to the NCU via X127. Log into Linux via WinSCP or Putty.<br>If the set NCU mode (no DHCP) is to be kept, the fixed IP address of the NCU must be changed.<br>Correctly enter the following entries into the file /user/system/etc/basesys.ini<br>[InternalInterface]<br>; With InternalIP and InternalNetMask (both must be set together),<br>; you can change the address on the internal/TCU/automation/system net.<br>InternalIP=192.168.214.1<br>InternalNetMask=255.255.255.0<br>Then restart the NCU. |
| | **Error cause 4.2** |
| | The NCU has a valid IP address in the system network, however, not the required IP address.<br>The NCU does not participate in DHCP operation and operates with a permanently set target address that is different from that saved in MMC.INI in the system network - however, this is a valid address. This is a configuration error. |
| | **Remedy 4.2**<br>The same as for remedy 4.1 |
| | **Error cause 4.3** |
| | The NCU has a valid IP address in the system network, however, not the required IP address.<br>The NCU participates in DHCP operation and was not able to obtain the corresponding required IP with the target address saved in the MMC.INI; this is the reason that it was dynamically allocated an address. A reason for this could be that the required IP was configured twice in the system network, or the required IP is not in the reserved range, but is located in the dynamic address range. This is a configuration error. |
| | **Remedy 4.3**<br>The same as for remedy 4.1 |
| | **Error cause 4.4** |
| | The NCU has a valid IP address in the system network, however, not the required IP address.<br>The NCU participates in DHCP operation as DHCP client and has a required IP that has been set but which does not match the system network. For instance, this situation occurs if a network address was used in the system network other than the default network address, but it was forgotten to change over the required IP of the NCU to this address range. The NCU is allocated an IP address that matches the system network from the dynamically allocated range, but which is different from its 'old' required IP that was not changed over. Up to and including SW 2.4.1, in this situation, the NCU cannot be addressed in the system network. The reason for this is that it has internally established its firewall regarding the required IP that was entered for it, which does not match the system network - which is why it does not respond at X120. |
| | **Remedy 4.4**<br>The same as for remedy 4.1 |
| HMI displays alarm 120202 "Wait for connection to NC/PLC" and/or no values (only '#'). | |
| | **Error cause 5.1** |

| Error | Possible error causes and remedies |
|---|---|
| | There is no physical connection.<br>Possible error causes include:<br>• A switch is not closed/has failed; this can also be the switch of an MCP<br>• The cable connection has become loose (poor contact)<br>• An incorrect cable was used (crossover/not crossover)<br>• A cable being used as data link is defective |
| | **Remedy 5.1** |
| | Establish a physical connection. |

### 14.6.2.4 TCU cannot establish a network connection to the HMI

| Error | Possible error causes and remedies |
|---|---|
| When booting, TCU signals 'ERROR: Network connection not established'. | |
| | **Cause of the error** |
| | There is no physical connection.<br>Notice: The display only refers to the situation where the direct connection of the TCU to the next connection nodes has not been established, i.e.<br>• The Ethernet cable is not directly inserted at the TCU,<br>• The Ethernet cable is not inserted at the direct connection partner or the partner is not available (e.g. MCP),<br>• The Ethernet cable at the TCU is defective. |
| | **Remedy** |
| | Establish the physical network connection (check the plug-in connections, replace the cable, check the connection partner).<br>Once the connection has been established, the TCU continues to boot. |
| When the TCU boots, it signals 'connection established – no DHCPServer available'. | |
| | **Cause of the error** |
| | The physical network connection does not exist after the next connection node to the system network. There is no physical connection from the TCU to the computer with the active DHCP server. The cables are either defective, have not been inserted or switches have not been closed. Switches can also be located on the MCP or distribution box. |
| | **Remedy** |
| | Establish the physical network connection (check the plug-in connections, replace the cable, check the connection partner).<br>Once the connection has been established, the TCU continues to boot. |
| When the TCU boots, it signals 'connection established – no DHCPServer available'. There is a physical network connection to the system network. | |
| | **Cause of the error** |

| Error | Possible error causes and remedies |
|---|---|
| | No active DHCP server is found in the system network.<br>Possible error causes include:<br>• Only a single computer has been configured in the system network with active DHCP server - and this computer is not available. The system network is not operated with synchronizing DHCP servers.<br>• No computer has been configured in the system network with active DHCP server. DHCP operation is disabled on all potential computers (NCU, PCU). This is a configuration error.<br>• In the system network, there is no NCU or PCU computer available that has DHCP operation activated on it and could be a server. |
| | **Remedy** |
| | In the system network, configure a computer that is available as DHCP server. |
| After it has booted, the TCU displays 'Waiting for HMI'. (The TCU cannot establish its connection to the HMI.) | |
| | **Error cause 1** |
| | The connection to the HMI has not been established - neither the connection configured in config.ini nor the default connection. An incorrect address for the HMI connection after booting has been set in the config.ini file. |
| | **Remedy** |
| | Correct the configuration using the "System Network Center" program. The TCU automatically reboots. |
| | **Error cause 2** |
| | The connection to the HMI has not been established - neither the connection configured in config.ini nor the default connection. The computer, to which the TCU should connect after booting according to the configuration in config.ini, cannot be reached because the computer is not switched on/is not available. |
| | **Remedy** |
| | Switch off the computer (NCU or PCU) and switch on again. |
| | **Error cause 3** |
| | The connection to the HMI has not been established - neither the connection configured in config.ini nor the default connection. The HMI on the computer, to which the TCU should connect after booting according to the configuration in config.ini, cannot be reached because the HMI is deactivated. |
| | **Remedy** |
| | Activate the HMI on the NCU using the service command sc enable HMI. |
| | **Error cause 4** |
| | The connection to the HMI has not been established - neither the connection configured in config.ini nor the default connection. The HMI on the computer, to which the TCU should connect after booting according to the configuration in config.ini, cannot be reached because the HMI has either not start or has crashed due to a software error. |
| | **Remedy** |
| | Switch off the computer (NCU or PCU) and switch on again. |

### 14.6.2.5 The TCU cannot establish a network connection to the assigned PCU

| Error | Possible error causes and remedies |
|---|---|
| The PCU does not have the IP address specified in the configuration, but an address from the IP range of the system network that can be dynamically allocated. In this error case, the TCU cannot reach its configured PCU for display. | |
| | **Error cause 1** |
| | IP address assigned twice<br>The PCU has a default IP address which is also configured for another NCU/PCU. This means that an IP address has been assigned twice and therefore there is a system configuration error.<br>The PCU is not an active DHCP server (otherwise, it would have the required IP), however, it is operated in the synchronizing DHCP mode. As the address requested from the PCU has already been assigned, the PCU is allocated an address from the dynamic address range via DHCP. |
| | **Remedy** |
| | A decision has to be made which computer should keep the IP address that has been assigned twice. If it is the PCU, then<br>• on the PCU, under network settings, this address should be again set as fixed IP address in the system network,<br>• at the NCU/PCU which runs in the DHCP synchronization mode ON_MASTER, a 'Clear DHCP' should be executed (for an NCU sc command sc clear dhcp –X120, for a PCU using the "System Network Center" program).<br>• at the computer, which uses the IP address assigned twice as second computer, the IP address should be re-configured. Notice: If this is a PCU, then it should be linked to this new address using the "System Network Center" program (this is the first screen in the program).<br>If the PCU is to change its address, then on the PCU<br>• under network settings, the new address should be set as fixed IP address in the system network,<br>• linked to this new address using the "System Network Center" program (first screen in the program)<br>• at the NCU/PCU, which runs in the DHCP synchronization mode ON_MASTER, a 'Clear DHCP' should be executed (for an NCU using the sc command, for a PCU, using the "System Network Center" program).<br>Restart the system. |
| | **Error cause 2** |
| | This was preceded by replacing the PCU hardware.<br>As a result of the hardware replacement, in spite of the same settings of the new PCU, the requested (set) IP address is still internally reserved for the previous hardware as this is an inherent feature of the system. The PCU is either active or non-active DHCP server, however, is operated in the synchronizing DHCP mode. As the address requested from the PCU is still reserved, the PCU is allocated an address from the dynamic address range via DHCP. |
| | **Remedy** |

| Error | Possible error causes and remedies |
|---|---|
| | Perform the following steps:<br>• Under network settings, on the PCU, set the required address again as fixed IP address in the system network.<br>• At the NCU/PCU, which runs in the DHCP synchronization mode ON_MASTER or is active DHCP server, execute a 'Clear DHCP' (for an NCU using the sc command sc clear dhcp – X120, for a PCU, using the "System Network Center" program).<br>• Restart the system. |
| | **Error cause 3** |
| | An address that does not belong to the network has been configured for the PCU.<br>An address has been configured for the PCU, which does not match the network address in the system network, and is therefore dynamically assigned an IP address from the active server via DHCP. The PCU is not an active DHCP server, but is operated in the synchronizing DHCP mode. |
| | **Remedy** |
| | Perform the following steps:<br>• Under network settings, on the PCU, set the required address again as fixed IP address in the system network.<br>• On the PCU, using the "System Network Center" program, link the DHCP service to this new address (first screen in the "System Network Center" program), if required, adapt the other DHCP settings (dynamic address range).<br>• At the NCU/PCU, which runs in the DHCP synchronization mode ON_MASTER or is active DHCP server, execute a 'Clear DHCP' (for an NCU using the sc command sc clear dhcp – X120, for a PCU, using the "System Network Center" program).<br>• Restart the system. |

## 14.6.3 Configuring TCP/IP diagnostics

The configuration of the network adapter is displayed in the "TCP/IP Configuration" window.

**System network**

On the system network, process data communication and image transmission of the components are executed with the operating software for the display units (TCUs).

A TCU and NCU are connected to the system network via Ethernet interface X120.

A PCU is connected via the ETH2 Ethernet adapter.

**Company network**

The company network is used, for example, to access the network drives.

An NCU is connected to the company network via Ethernet interface X130.

A PCU is connected via the ETH1 Ethernet adapter.

### Availability of the network connections

#### Network adapter connection



White Network cable inserted



Red Network cable not inserted

#### Availability

The availability describes the percentage of faulty packages compared to all the sent and received packages.

Problems in the company network (e.g. logical drives that cannot be accessed, double IP address, etc.) as well as settling time during power up can result in fluctuations in availability.



Green Greater than 95%



Yellow 50 - 95%



Red Less than 50%

### Details of the network connections

- **Computer name**
- **DNS name**
  Computer name of the controller as stored at the DNS server (**D**omain **N**ame **S**ervice).
- **MAC address**
  Physical address of the network adapter

- **Address type**
  Information about the configuration of the network adapter in the configuration file "basesys.ini":
  - DHCP: DHCP is active for this network adapter.
    The information behind the hyphen "-" specifies the DHCP mode of operation:
    *off* - DHCP is deactivated for this adapter
    *Client* - a DHCP client that receives an IP address and further data from a server, is running on the interface
    *Server* - the NCU provides a DHCP server on this interface that supplies clients on this network with IPs
    *Synced Server* - a protocol is active with which several NCUs synchronize with each other. This ensures that always only one of these NCUs functions as DHCP server and, for example, no double IP addresses
    are assigned.
    If a network adapter operates as "Synced Server", the "State of sync. DHCP server" line displays additional information.
    If no changes have been made for the network adapters in the configuration file, then "Default" is also output.
  - Manual
    The settings IP address, subnet mask, DNS server 1, DNS server 2 and gateway are configured in the "basesys.ini".
    Note:
    In the "Change" mode, either "Manual" or "DHCP" can be selected (only for company network, X130).
- **IP address assigned**
  Current IP address of the network adapter.
- **Subnet mask assigned**
- **DHCP server**
  IP address of the network adapter (for address type "DHCP").
- **State, DHCP server**
  State of the synch. DHCP server:
  - Active (actual NCU is active in the group of all NCUs in the "Synced Server" mode and transfers the IP addresses);
  - Standby (NCU is not active, if the active server fails, this NCU could take over the task of the server).
- **Mode, DHCP synchron.**
  Priority of the synch. DHCP server:
  Low, High, Master
- **DNS server 1, DNS server 2**
  Domain Name Server
- **Gateway**
- Note
  All information that is not available is marked in the relevant table line with a hyphen "-".

### Modifiable parameters

The following parameters of the company network adapter (X130) can be set by selecting the "Address type":

- Address type
- IP address
- Subnet mask
- DNS server 1 and 2
- Gateway (applies for system and company network)

### Address types

- DHCP
  The values for the settable parameters IP address, subnet mask, DNS server 1, DNS server 2 and gateway are defined automatically by the DHCP server, in accordance with the entry in the "basesys.ini".
- Manual
  The values for the settable parameters IP address, subnet mask, DNS server 1, DNS server 2 and gateway can be specified manually; these are then stored in the "basesys.ini".

## References

For additional information about network configuration, please refer to the following documentation:

SINUMERIK 840D sl, Operator Components and Networking (IM5)

## Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key and the "Bus TCP/IP" softkey.







3. Press the "TCP/IP diagnostics" and TCP/IP Config." softkeys.
   The "TCP/IP Configuration" window opens.







4. Press the "Modify" softkey to change specific parameters.
   The fields that can be edited have a white background.




5. Enter the desired changes and press the "OK" softkey to confirm the entries.




6. Press the "New display" softkey to refresh the display.

## 14.6.4 Ethernet network diagnostics

### 14.6.4.1 Diagnostics of the network and DHCP server

In order to more effectively influence the commissioning, press the "Network diagnostics" softkey to easily represent the data and availability of each network node on the Ethernet. The results display of this scan process delivers, among other things, the following information:

- Component identification of the hardware
- TCP/IP address / MAC address
- DHCP server detection
- Status of the nodes

This gives you the option to test whether the current topology matches the reference topology.

### Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key and the "Bus TCP/IP" softkey.







3. Press the "TCP/IP diagnostics" softkey.
   The "TCP/IP Diagnostics" window opens and displays the current availability of the network connections.




4. Press the "Network diagnostics" softkey to browse your networks.
   The "Network Status" window opens and displays the addresses of the HMI (X120) system network, the PN (X150) system network and the (X130) company network.
   The corresponding fields in the window remain empty for non-accessible networks.




5. Press the "New display" softkey to refresh the display.

   - OR -




Press the "DHCP server" softkey to fetch information and query the status of the DHCP server.

The "DHCP Server" window opens.

For each active DHCP server, the status, the role (priority), the MAC address, the TCU version and the lease version are displayed in tabular form in the upper part of the window. The required IP with the associated address area appears in the lower part of the window.

Any differing data is displayed in red.




6. Press the "Address list" softkey to acquire all information about the active DHCP server.

   The "Address list" window is opened.

   The IP address, the client ID in the form of the MAC address or the CF card number, the lease version, the device number, the DNS name and additional information are displayed for each node in a tabular form.

   Inactive nodes are grayed out.

### 14.6.4.2 Messages

All the system-relevant state and error messages from the "syslog" log file are displayed via the "Messages" softkey.

### Procedure




1. The "Network Status" window opens.




2. Press the "Messages" softkey.

   The "Messages" window opens.




3. Press the "Show all", "DHCP", "MCP" or "TCU" softkey to filter the display.










### 14.6.4.3 Network trace

If network problems occur, you can record the data traffic as a log file for analysis and evaluation via the "Network Trace" window.

## General information

- The function is protected with the standard password NETWORK

  **Note**

  Change the standard password to prevent improper use.

- The log is recorded via the tcpdump program (Linux) or the windump program (Windows). Further information can be found at www.tcpdump.org

## Requirement

If you want to change the password, you require access level 1 (manufacturer).

## Procedure




1. The "Network Status" window is open.




2. Press the "Network trace" softkey.
   The "Set Password" window opens.




3. Enter the password and press the "OK" softkey to confirm the entry.
   The "Network Trace" window opens.




4. If you have stored parameters in a file, press the "From file" softkey in order, for example, to read-in the file from a USB.
   The "Select File" window opens.




5. Select the file from a directory and confirm with "OK".

   - OR -

4. If required, parameterize the recording conditions in the "Parameters (tcpdump)" input field.




5. Press the "Start" softkey to record the log.
   **Note:** Recording stops automatically after 20 minutes.




6. Press the "Stop" softkey to complete the recording.
   The "Export Trace File" window opens.




7. Select a storage location, assign a file name and confirm with "OK".
   The trace file is moved to the storage location. With "Cancel", the log remains in the following directory:
   HMI data/Logs/Network trace/tcpdump.pcap or windump.pcap




8. If the log already exists, press the "Export trace file" softkey to save the current log file, for example, on a USB.
   The "Select Storage Location" window opens.




9. Select a storage location, assign a file name and confirm with "OK".

### Change password







1. The "Network Trace" window is open.
2. Press the "Change password" softkey.
   The "Change Password" window opens.
   Enter a new password and confirm with "OK".

## Trace parameters

You can parameterize the conditions for the log recording via the "Parameters (tcpdump)" input field in the Network Trace window.

## Syntax

```
<switch> <option>
```

## <switch> parameters

| Value | Meaning |
|---|---|
| `-D` | Displays a list of all network interfaces. All other <Switch> parameters are ignored. |
| `-i <interface>` | Defines which network interface has to be checked. |
| `-n` | Host names are not resolved. |
| `-c <number>` | The recording is stopped after a defined number of packets. |
| `-C <file size>` | Limits the file size. When the file size is reached, a new file with name and consecutive number, starting with 1, is created. The unit of the file size corresponds to million bytes. |
| `-e` | Outputs the Ethernet header. |
| `-q` | Creates an abbreviated log version. |
| `-vv` | Creates more detailed packet information. |
| `-S` | Changes the packet sequence number from absolute to relative. |
| `-A` | Outputs the contents of a packet as ASCII. |
| `-xx` | Outputs the contents of a packet as Hex. |
| `-XX` | Outputs the contents of a packet as Hex and as ASCII. |
| `-s <bytes>` | Specifies the packet size in bytes that has to be recorded per packet. The default is 68 bytes. |

**Note**

The value `-w` is always implicitly contained in the call for the creation of the standard log file:

HMI data/Logs/Network trace/tcpdump.pcap or windump.pcap

### Parameter <option>

| Option | Value | Meaning |
|---|---|---|
| `host` | IP address or host name | Only those packets are recorded that have the specified IP address or host name as source or as destination. |
| `net` | Network range | Only those packets are recorded that have an IP address or host name from the specified network range as source or as destination. |
| `port` | A number from 0 to 65535 | Records packets that have the specified port, e.g. 5900, on the source or on the destination side. |
| `portrange` | A range between 0 and 65535 | Records packets that have a port in the specified range on the source or on the destination side. |
| `src` | host, net, port or portrange | Only those packets are recorded that have the specified value as source. |
| `dst` | host, net, port or portrange | Only those packets are recorded that have the specified value as destination. |
| `proto` | ether, fddi, tr, wlan, ip, ip6, arp, rarp, decnet, tcp and udp | Network protocol in which packets can be recorded. |

You can link several options using the following expressions:

`and`, `or`, `not`

You can limit extended expressions using the following characters:

`( )` and `'`

### Reference

Further parameter assignments can be found at www.tcpdump.org

### Examples

```
-n port 22                          ; Records whether packets
                                    arrive at port 22 of the local
                                    system.
-i eth0 port 4080                   ; Records all data for port
                                    4080 arriving at the eth0
                                    interface.
-i 2                                ; Records all data arriving at
                                    the interface with adapter
                                    number 2.
-n host 10.113.20.0                 ; Records the network traffic
                                    from system 10.113.20.0,
                                    whereby the IP address is not
                                    changed to the host name.
```

```
192.168.214.4 and not net 10.113.20.0 ; Records the entire network
                                        traffic of system
                                        192.168.214.4 which does not
                                        run via system 10.113.20.0.
-i 10.113.20.0 -q '(tcp port 80) or   ; Records HTTP and also HTTPS
(tcp port 443)'                         packets of the network
                                        interface from system
                                        10.113.20.0.
```

### 14.6.4.4 Accessible nodes

By actuating the "Accessible nodes" softkey in the network, the list of accessible nodes on X120 and, if necessary, on X150, is created and displayed. The list content can be compared as current topology with a reference topology.

**Note**

Although handheld terminals and EKS identification systems are not determined for the list of accessible nodes, they can be added as additional nodes to the topology.

## Procedure




1. The "Network Status" window opens.




2. Press the "Accessible nodes" softkey.
   The "Accessible Nodes" window opens.
   The IP address, HW-Id/type, status and location are displayed in tabular form for each node.




3. Press the "Sort" softkey.













4. Press the "By name", "By IP", "By type" or "By location" softkeys to sort the display.

- OR -




3. Press the "Topology" softkey if you require an overview of all of the networked nodes, e.g. to check if there are connection problems.

   The "Network Topology" window opens.

   The interface, port and the actual connection are displayed for each node using a tree topology.

   **Note:** The station name must be defined for all nodes so that the topology is correctly displayed.







4. Press the "Expand all" or "Close all" softkey to maximize or minimize the tree topology.




5. Press the "Follow connection" softkey to follow nodes in the structure tree using the marked line.

   - OR -




3. If further information is provided for a selected node, press the "Details" softkey to open the Web user interface.

   **Note:** The installed operating software must be located in the same network as the device to be displayed. If necessary, this is achieved by the assignment of a second IP address (alias).

   The "Details" window opens.







4. Press the "Zoom +" or "Zoom -" softkey until the desired display size is reached.

### Additional nodes







3. Press the "Additional nodes" softkey to search for nodes that are not listed under "Accessible nodes".

   The "Search for Additional Nodes" window opens. If necessary, change the address range and confirm with "OK".

   The found nodes are appended in the table; the HW-Id/type, Location, Contact and Description fields can then be edited and, by activating the checkbox, added to the first column, and, with a subsequent save, added to the reference topology.

### Comparing







3. Press the "Compare" softkey to compare the current list of nodes either with the previously saved manufacturer topology, a reference topology or a saved topology in the .xml format.

   The "Compare with" window opens. Position the cursor on the desired option box and confirm with "OK".

Deviations during the comparison are identified by being color-highlighted:

- Red: missing or non-accessible components compared with the reference topology.
- Pink: components that deviate in their attribute values compared with the reference topology.
- Blue: new components not contained in the reference topology.

### Saving




3. Press the "Save" softkey to save the current topology.
   The "Save as" window opens.




4. Position the cursor on the desired option box, specify a file name and possibly a comment, and confirm with "OK".

**Note**

Manufacturer topology and reference topology can be created with access level 3 (user) or higher and are always saved on the active DHCP server.

Default storage locations:

- Current topology (.xml or .csv):
  The following storage locations are offered depending on the configuration:
  - Network topology (storage location: HMI-data / data backups)
  - Local drive
  - Network drives
  - USB

  In CSV format, ";" is preset as a separator; it can be changed in sldgconfig.xml
- Reference topology:
  /user(_base)/common/tcu/$soll
- Manufacturer topology:
  /oem(_base)/common/tcu/$soll

# Service and diagnostics 15

## 15.1 NC/PLC variables

### 15.1.1 Displaying and editing PLC and NC variables

Changes can only be made to the NC/PLC variables with the appropriate password.

| ⚠ WARNING |
| --- |
| **Incorrect parameterization**<br>Changes in the states of NC/PLC variables have a considerable influence on the machine. Incorrect configuration of the parameters can endanger life and cause damage to the machine. |

In the "NC/PLC Variables" window, enter the NC system variables and PLC variables that you want to monitor or change in the list:

- Variable
  Address for NC/PLC variable.
  Incorrect variables have a red background and are displayed with a # character in the value column.
- Comment
  Any comment on the variables.
  The columns can be displayed and hidden.
- Format
  Specify the format in which the variable will be displayed.
  The format can be specified (e.g. floating point).
- Value
  Displays the actual value of the NC/PLC variables.

| PLC variables | |
| --- | --- |
| Inputs | • Input bit (Ex), input byte (EBx), input word (EWx), input double word (EDx)<br>• Input bit (Ix), input byte (IBx), input word (IWx), input double word (IDx) |
| Outputs | • Output bit (Ax), output byte (ABx), output word (AWx), output double word (ADx)<br>• Output bit (Qx), output byte (QBx), output word (QWx), output double word (QDx) |
| Bit memory | Memory bit (Mx), memory byte (MBx), memory word (MWx), memory double word (MDx) |
| Times | Time (Tx) |

| PLC variables | |
|---|---|
| Counters | • Counter (Cx)<br>• Counter (Cx) |
| Data | • Data block (DBx): Data bit (DBXx), data byte (DBBx), data word (DBWx), data double word (DBDx)<br>• Data block (VBx): Data bit (VBXx), data byte (VBBx), data word (VBWx), data double word (VBDx) |

| Formats | |
|---|---|
| B | Binary |
| H | Hexadecimal |
| D | Decimal without sign |
| +/-D | Decimal with sign |
| F | Floating point (for double words) |
| A | ASCII character |

## Notation examples

Permissible notation for variables:

- PLC variables: EB2, A1.2, DB2.DBW2, VB32000002
- NC variables:
  - NC system variables: Notation $AA_IM[1]
  - User variables / GUD: Notation GUD/MyVariable[1,3]
  - OPI notation: /CHANNEL/PARAMETER/R[u1,2]

**Note**

If the PLC user program writes a string in an NC/PLC variable, the string will only be displayed correctly if the variable is parameterized as a field variable of the type "A" (ASCII) on the NC side.

### Example of a field variable

| Variable | Format |
|---|---|
| DBx.DBBy[<number>] | A |

## Inserting variables

The start value for "Filter/Search" of variables differs. For example, to insert the variable $R[0], enter the following start value:

- The start value is 0 if you filter according to "System variables".
- The start value is 1 if you filter according to "All (no filter)". In this case, all signals are displayed and shown in the OPI notation.

The GUD from the machine data is only displayed in the Search window for the variable selection when the associated definition file has been activated. Alternatively, enter the variable you are searching for manually, e.g. GUD/SYG_RM[1]

The following machine data is representative for all variable types (INT, BOOL, AXIS, CHAR, STRING): MD18660 $MN_MM_NUM_SYNACT_GUD_REAL[1].

**Note**

- System variables can be dependent on the channel. When the channel is switched over, the values from the selected channel are displayed.
  You have the option of having the variable displayed for a specific channel, e.g. $R1:CHAN1 and $R1:CHAN2. The values of channel 1 and channel 2 are displayed, irrespective of the channel you are in.
- For user variables (GUD), it is not necessary to specify whether they are global or channel-specific GUD. The first element of a GUD array starts with index 0 as for NC variables.
- Use the tooltip to show the OPI notation for NC system variables (except for GUD).

**Servo variables**

Servo variables can only be selected and displayed at "Diagnostics" → "Trace".

## Changing and deleting values




1. Select the "Diagnostics" operating area.




2. Press the "NC/PLC variables" softkey.

- OR -




1. Select the "Startup" operating area.




2. Press the "PLC" and "NC/PLC variab." softkeys.




The "NC/PLC Variables" window opens.

3. Position the cursor in the "Variable" column and enter the required variable.




4. Press the <INPUT> key.
   The operand is displayed with the value.




5. Press the "Details" softkey.
   The "NC/PLC Variables: Details" window opens. The information for "Variable", "Comment" and "Value" is displayed in full length.




6. Position the cursor in the "Format" field and select the required format with <SELECT>.




7. Press the "Display comments" softkey.
   The "Comments" column is displayed. You have the option of creating comments or editing existing comments.




Press the "Display comments" softkey once again to hide the column.




8. Press the "Change" softkey if you would like to edit the value.
   The "Value" column can be edited.




9. Press the "Insert variable" softkey if you wish to select a variable from a list of all existing variables and insert this.
   The "Select Variable" window opens.




10. Press the "Filter/search" softkey to restrict the display of variables (e.g. to mode group variables) using the "Filter" selection box and/or select the desired variable using the "Search" input box.




11. Press the "Delete all" softkey if you would like to delete all the entries for the operands.




12. Press the "OK" softkey to confirm the changes or the deletion.

- OR -




Press the "Cancel" softkey to cancel the changes.

## Editing a variable list

You can edit the variable list using the "Insert line" and "Delete line" softkeys.




When you press the softkey, a new line is inserted before the line marked by the cursor.

You can only use the "Insert line" softkey if there is at least one empty line at the end of the variable list.

The softkey is deactivated if there is no empty line.




When you press the "Delete line" softkey, the line marked by the cursor is deleted.

An empty line will be added at the bottom of the variable list.

## Changing operands

Depending on the type of operand, you increment or decrement the address or address index by one place at a time using the "Operand +" and "Operand -" softkeys.

**Note**

**Axis name as index**

If the axis name is the index, the "Operand +" and "Operand -" softkeys have no effect, e.g. $AA_IM[X1].

**Examples**

Operand +

DB97.DBX2.5
Result: DB97.DBX2.6
$AA_IM[1]
Result: $AA_IM[2]

Operand –

MB201
Result: MB200
/Channel/Parameter/R[u1,3]
Result: /Channel/Parameter/R[u1,2]

## 15.1.2 Saving and loading screen forms

You have the option of saving the configurations of the variables made in the "NC/PLC variables" window in a screen form that you reload again when required.

### Editing screen forms

If you change a screen form that has been loaded, then this is marked using with * after the screen form name.

The name of a screen form is kept in the display after switching-off.

### Procedure

1. You have entered values for the desired variables in the "NC/PLC variables" window.
2. 
   Press the ">>" softkey.
3. 
   Press the "Save screen" softkey.
   The "Save screen: Select archiving" window opens.
4. 
   Position the cursor on the template folder for variable screen forms in which your actual screen form should be saved and press the "OK" softkey.
   The "Save screen: Name" window opens.
5. 
   Enter the name for the file and press the "OK" softkey.
   A message in the status line informs you that the screen form was saved in the specified folder.
   If a file with the same name already exists, they you will receive a prompt.
6. 
   Press the "Load screen" softkey.
   The "Load screen" window opens and displays the sample folder for the variable screen forms.
7. Select the desired file and press the "OK" softkey.
   You return to the variable view. The list of all of the predefined NC and PLC variables is displayed.

## 15.2 Displaying the Service overview

The operating state of machine axes and drives is displayed in the "Service Overview" window.

You configure the display via the "Extended selection >" softkey or directly via the drop-down list according to the following criteria:

- All NC axes and drives (= default)
- NC axes
- Drives without NC axis assignment

### Enable displays

- On/Off1
- Off2
- Off3
- From the drive: Operation enabled
- From the infeed: Enable operation
- Enable pulses
- Speed controller enable NC
- Enable pulses
- Drive ready
- Heat sink temperature
- Power section in $i^2t$ limiting
- Motor temperature
- **Measuring system 1 active / measuring system 2 active**
  The states for measuring system 1/2 have the following meaning:

| Symbol | Meaning |
|---|---|
|  | The position measuring system is active. |
|  | The position measuring system is parked.<br>- OR: -<br>The position measuring system is not configured. |
|  | The position measuring system is passive. |

### Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key and the “Axis diag" softkey.




3. Select an overview using the "Select" field.
   - OR -
   Press the "Go to selection" softkey to select an overview.




## 15.2.1 Selecting axes and drives

To display certain enable signals and statuses of machine axes, you can arrange a selection of all the axes and drives that are available in any sequence you wish.

**Note**

The following selection configurations have been previously defined, and cannot be changed or deleted:

- All NC axes and drives
- NC axes
- Drives without NC axis assignment

### Selection overview

The following information is provided in the selection overview.

| Column | Description |
|---|---|
| Bus No. | Bus number for drives without NC assignment.<br>No entry for NC axes. |
| Slave address | Slave address for drives without NC assignment.<br>No entry for NC axes. |
| Device No. | Device number for drives without NC assignment.<br>No entry for NC axes. |
| Drive object No. | Drive object number for drives without NC assignment.<br>No entry for NC axes. |

| Column | Description |
|---|---|
| Name/identifier | For NC axes contains the standard machine axis identifier and the user-defined machine axis names.<br>For drives without NC assignment, contains the drive object names. |
| Assignment | NC is displayed for an internal object assignment.<br>PLC is displayed for an external object assignment. |

**Note**

The selection overview differs in the information provided regarding the slave address and device number between PROFIBUS and PROFINET configuration.

## Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key and the “Axis diag" softkey.
   The "Service Overview" window opens.







3. Press the "Extended selection" softkey.
   The window "Extended selection: Axes and drives" opens.




4. Press the "New selection" softkey.
   The "New selection" dialog is opened.
5. Enter a display and a file name in the entry fields.




6. Confirm the entry with the "OK" softkey.
   The "Edit selection" window is opened for the newly created selection.
7. Set checkmarks in the check boxes for the axes and drives that you wish to display.
   - OR -
   Press one of the following softkeys to obtain a certain selection:




All axes/drives are selected.




No axis/drive is selected.




Only internal drives are selected.




Only external drives are selected.




Those axes are selected that are assigned to a real drive.




All axes are selected, which are defined in a minimum of one channel.




8. Press the "OK" softkey to confirm your selection.




9. Press the "Delete selection" softkey to delete the actual selection.




10. Press the "Edit selection" softkey to edit the actual selection.
    The "Edit selection" window opens.




11. If you click the "Back" softkey, you will return to the "Service Overview" main screen.

## 15.2.2 Axis diagnostics

### Application

The following information is displayed in the "Service Overview" window:

- Check of the setpoint branch (e.g. programmed position setpoint, speed setpoint, spindle speed setpoint)
- Check of the actual value branch (e.g. actual position value, measuring system 1 or 2, actual speed value)
- Optimization of the position control loop of the axis (e.g. following error, control deviation, servo gain factor)
- Check of the entire control loop of the axis (e.g. through position setpoint/actual-value comparison and speed setpoint/actual-value comparison)
- Check of hardware errors (e.g. by checking the encoder: If the axis is moved mechanically, the actual position value must change)
- Setting and check of the axis monitoring functions.

### Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key and the “Axis diagnostics" softkey.
   The "Service Overview" window opens.







3. Press the "Service axis" softkey.
   The "Service Axis/Spindle" window opens.




4. Press the "Axis +" or "Axis -" softkey to scroll the axes forward or backward.




- OR -




Press the "Axis selection" softkey.

The "Axis Direct Selection" window opens.

Select the required axis directly from those available in the drop-down list box.




6. Confirm the selection with "OK".

The values of the axis are displayed.

## Display data

| Display data | Meaning |
|---|---|
| Following error | The difference between the position setpoint and the actual position value of the active measuring system 1 or 2.<br>Unit: mm, inch or degrees |
| System deviation | The difference between the position setpoint at the position controller input and the actual position value of the active measuring system 1 or 2.<br>Unit: mm, inch or degrees |
| Contour deviation (axial) | The actual contour deviation is displayed with this value (variations of the following error caused by equalization operations on the speed controller due to load changes).<br>The contour deviation results from the difference between an actual position pre-calculated from the position setpoint and the actual position value of active measuring system 1 or 2.<br>Unit: mm, inch or degrees |
| Servo gain factor (calculated) | The servo gain factor in the display is calculated by the NC according to the following equation:<br>$K_v$ - factor = $\frac{\text{Speed setpoint}}{\text{Following error}}$ ;<br>Unit (for default setting): $\frac{[\text{m/min}]}{[\text{mm}]}$ ;<br>Velocity setpoint = setpoint currently being output to the axis/spindle<br>**References:**<br>Basic Functions Function Manual; Velocities, Setpoint / Actual Value Systems, Closed-Loop Control (G2) |
| Active measuring system | **0:** No measuring system active.<br>**1:** Measuring system 1 active.<br>**2:** Measuring system 2 active. |
| State of measuring system 1<br>State of measuring system 2 | The state of measuring system 1 or 2 is output here:<br>Active - Parked - Passive |

| Display data | Meaning |
|---|---|
| Actual position value measuring system 1 | The actual position of the axis measured via measuring system 1 or 2.<br>The position is displayed in the machine coordinate system (no work offsets or tool offsets taken into account).<br>Unit: mm, inch or degrees |
| Actual position value measuring system 2 | |
| Position setpoint | Specified position transferred from the interpolator to the position control<br>Unit: mm, inch or degrees |
| Absolute compensation value measuring system 1 | Display of the absolute compensation value for measuring system 1 or 2.<br>The compensation value consists of the sum of backlash and leadscrew error compensation for the actual axis position.<br>Unit: mm, inch or degrees |
| Absolute compensation value measuring system 2 | |
| Compensation, sag + temperature | Display of the compensation value calculated for the current axis position based on the total of the sag and temperature compensations.<br>Unit: mm, inch or degrees |
| Actual speed value, active encoder | The pulses supplied by the encoder are evaluated by the NC and displayed.<br>Unit: %<br>100% means maximum speed. |
| Speed setpoint, drive | Speed setpoint transferred to the drive (= speed setpoint from position controller and feedforward control).<br>Unit: %<br>100% means maximum speed setpoint. |
| Programmed spindle speed setpoint | Speed setpoint programmed by the user.<br>Unit: rpm<br>e. g.: Input: S1000; display: 1000 rpm<br>Display applies to spindles only. |
| Spindle speed setpoint current | Current active speed setpoint with correct sign, including calculated compensation value and any active speed limitation (specified by setting or machine data)<br>Unit: rpm<br>Display applies to spindles only. |
| Position offset to the leading axis / spindle actual value | The currently valid position offset value is displayed here (relative to the actual value) if a position offset (angular offset between the following and leading spindle) has been programmed within the synchronous spindle functionality.<br>Unit: mm, inches, degrees<br>**References:**<br>Function Manual, Extended Functions; Synchronous Spindle (S3) |
| Position offset to the leading axis / spindle setpoint | The currently valid position offset value is displayed here (relative to the setpoint) if a position offset (angular offset between the following and leading spindle) has been programmed within the synchronous spindle functionality.<br>Unit: mm, inches, degrees<br>**References:**<br>Function Manual, Extended Functions; Synchronous Spindle (S3) |

| Display data | Meaning |
|---|---|
| Override | The effective correction factor of the feed or spindle correction switch is displayed.<br>Unit: % |
| Current gear stage | Display of the current actual gear stage<br>With axes, this is only displayed if a spindle is assigned to the axis. The display corresponds to the NC/PLC interface signal:<br>DB31, ... DBX16.0-2 (actual gear stage)<br>**References**:<br>Function Manual, Basic Functions; Spindles (S1) |
| Parameter set (axis) | Displays which of the six parameter sets of the position controller is active.<br>**References**:<br>Basic Functions Function Manual; Velocities, Setpoint / Actual Value Systems, Closed-Loop Control (G2) |
| Controller mode | Display of the current controller state:<br>• Position control<br>• Speed control<br>• Hold<br>• Parking<br>• Follow-up<br>• Braking<br>**References:**<br>Function Manual, Basic Functions; Various NC/PLC interface signals and functions (A2) |
| Feedforward control mode | Indicates whether and if so, which mode of the dynamic feedforward control for the axis is active:<br>• Inactive<br>• Velocity<br>The velocity-dependent speed feedforward control is active.<br>• Torque<br>The acceleration-dependent torque feedforward control is active (in combination with the speed feedforward control)<br>**References:**<br>Extended Functions Function Manual; Compensations (K3) |

| Display data | Meaning |
|---|---|
| "Referenced" state | Status display for reference point approach (axis):<br>• Curr. MS need not be referenced<br>(active measuring system does not require referencing)<br>• Curr. measuring system referenced<br>• Curr. measuring system must be referenced<br>(active measuring system requires referencing)<br>The display depends on the settings in the machine data:<br>• MD34110 $MA_REFP_CYCLE_NR<br>• MD20700 $MC_REFP_NC_START_LOCK<br>The display corresponds to the NC/PLC interface signal:<br>DB31, ... DBX60.4 and 60.5 (referenced/synchronized 1 or 2)<br>**References:**<br>Function Manual Basic Functions; Reference Point Travel (R1) |
| QEC state | Indicates whether and if so, which method of the quadrant error compensation (QEC) is active for the axis:<br>• Inactive<br>• Neural QEC learning active<br>• Conventional QEC active<br>• Conventional QEC w. adapt. Corr. v. active<br>(conventional QEC with adaptation of the compensation value active)<br>• Neural QEC active<br>• Neural QEC w. adapt. Measurement duration active<br>(neural QEC active with adaptation of the measuring time active)<br>• Neural QEC w. adapt. Decay time corr. v. active<br>• (neural QEC with adaptation of decay time of compensation value active)<br>• Neural QEC w. adapt. Meas. dur. + decay time Corr. v. active<br>(neural QEC active with adaptation of measuring time and decay time of compensation value active)<br>**References:**<br>Extended Functions Function Manual; Compensations (K3) |

| Display data | Meaning |
|---|---|
| "Travel to fixed stop" state | Indicates whether or not the axis has fulfilled the conditions for "Fixed stop reached" when the "Travel to fixed stop" function is active (IS DB31, ... DBX62.5):<br>• Normal control ("Travel to fixed stop" function not activated)<br>• Fixed stop reached<br>• Failed<br>**References:**<br>Function Manual, Basic Functions; Travel to Fixed Stop (F1) |
| Torque limitation value | Indicates the value programmed via FXST[x] or SD43510 $SA_FIXED_STOP_TORQUE or the value defined via MD37010 $MA_FIXED_STOP_TORQUE_DEF for the clamping torque for "Travel to fixed stop".<br>Unit: % of maximum torque<br>**References:**<br>Function Manual, Basic Functions; Travel to Fixed Stop (F1) |

## 15.2.3 Service drive

### Display drive and motor data

The "Service Drive" window displays important information about the status of the motors and drive modules, such as motor temperature and DC-link voltage.

### Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key and the “Axis diag" softkey.
   The "Service Overview" window opens.







3. Press the "Service drive" softkey.
   The "Service Drive" window opens.







4. Press the "Drive +" or "Drive -" softkey to scroll forward or backward.

- OR -




Press the "Drive selection" softkey.
The "Drive direct selection:" window opens.
Select the desired drive directly from the selection list.




5. Confirm the selection with "OK."
The drive data is displayed.

## Overview

The individual status displays, warnings, messages, etc. that are displayed in the "Service Drive" window are explained in the following.

Further information about the interface signals can be found in the following documentation: Function Manual, Basic Functions:

- Section "Various NC/PLC interface signals and functions (A2)"
- Chapter "NC/PLC interface signals (Z1)"

Further information about the drive parameters can be found in the following documentation: List Manual, SINAMICS S120/S150 (Drive Object SERVO)

## PLC pulse enable

The display, whether the pulse enable from the PLC is available for the drive corresponds to the interface signal:
DB31, ... DBX21.7 "Pulse enable".

| Status | Meaning | Display |
|---|---|---|
| 1 | Pulse enable for this drive is activated by the PLC. | Yes |
| 0 | The pulses for the drive module are disabled from the PLC. | No |

## Speed controller enable NC

The display, whether the speed controller enable from the NC is available for the drive corresponds to the interface signal:
DB31, ... DBX61.6 "Speed controller active".

| Status | Meaning | Display |
|---|---|---|
| 1 | Speed controller enable from the NC present. | Yes |
| 0 | Speed controller enable from the NC not present. | No |

### Rampup function generator quick stop

The status display for the ramp-function generator quick stop corresponds to the interface signal:
DB31, ... DBX92.1 "Ramp-function generator disable active".

| Status | Meaning | Display |
|---|---|---|
| 1 | Ramp-up function generator quick stop is active. The drive is stopped without a ramp function with speed setpoint = 0 and without pulse suppression. | Yes |
| 0 | Ramp-up function generator quick stop is not active for the drive. | No |

### Pulses enabled

The message whether the pulses have been enabled for the drive corresponds to the interface signal:
DB31, ... DBX93.7 "Pulses enabled".

| Status | Meaning | Display |
|---|---|---|
| 1 | The drive module pulses are enabled. The axis/spindle can now be traversed. | Yes |
| 0 | The drive module pulses are suppressed. The axis/spindle can therefore not be traversed. | No |

### Drive ready

The display of the current status of the selected drive corresponds to the interface signal:
DB31, ... DBX93.5 "Drive ready".

| Status | Meaning | Display |
|---|---|---|
| 1 | The drive is ready. | Yes |
| 0 | The drive is not ready. | No |

### Ramp-up phase

The display of the current ramp-up phase of the selected drive corresponds to the drive parameter:
r0002 "Drive operating display".

### Number of faulty signs-of-life

Display of communications errors detected in hardware between NC and drive.

**Note**

If the display shows a value other than "0", please contact your Siemens regional office!

### SC1 drive alarm message

Displays (yes/no) whether messages of status class 1 are active. Status class 1 messages are alarms with the following properties:

- They lead to internal responses (e.g. regenerative braking, immediate pulse suppression)
- They are modal.

This is a group message. For details on the actually pending drive alarms, please refer to the "Drive System Diagnostics" window, see Section "Displaying drive states (Page 296)."

### DC-link voltage, smoothed

The display of the smoothed actual value of the DC-link voltage of the selected drive corresponds to the drive parameter:
r0026 "DC-link voltage smoothed".

Unit: Volts

### Speed setpoint, smoothed

The display of the smoothed speed setpoint - measured according to the setpoint limitation for the P component of the speed controller - corresponds to the drive parameter:
r1438 "Speed controller speed setpoint".

Unit: rpm

### Actual speed value

The display of the smoothed actual value of the motor speed corresponds to the drive parameter:
r0021 "Actual speed value smoothed".

Unit: rpm

### Actual current value, smoothed

The display of the smoothed actual current value corresponds to the drive parameter:
r0078[1] "Actual current value, torque-generating".

Unit: A

### Motor temperature

The display of the current temperature in the motor corresponds to the drive parameter:
r0035 "Motor temperature"
Unit: °C

### Integrator disabling

The display as to whether the integrator of the speed controller is active, corresponds to the interface signal:
DB31, ... DBX93.6 "n-controller integrator disabled".

| Status | Meaning | Display |
|---|---|---|
| 1 | The requested shutdown of the speed controller integrator is active in the drive. The speed controller has been switched over from the PI to P control response. | Yes |
| 0 | The integrator of the speed controller is enabled. The speed controller functions as a PI controller. | No |

### Parking axis

Displays (yes/no) whether it is a parking axis/spindle.

**Note**

With parking axes/spindles, all encoder-specific monitoring and evaluation functions are switched off. This allows the encoder to be withdrawn without initiating an alarm.

### Specified drive data set

The display (default: DDS0) of which of the eight drive parameter sets is to be activated by the PLC corresponds to the interface signal:
DB31, ... DBX21.0 to 21.2 "Drive parameter set selection A, B, C."

More detailed information on working with the drive and motor data sets can be found in:
Commissioning Manual, IBN CNC: NCK, PLC, Drive

### Actual drive data set

The display (default: DDS0) of which of the eight drive parameter sets is currently active corresponds to the interface signal:
DB31, ... DBX93.0 to 93.2 "Active drive parameter set A, B, C."

### Specified motor data set

The display (MDS0...3) of which motor data set is to be activated by the PLC corresponds to the interface signal:
DB31, ... DBX21.3 to 21.4 "Motor selection A, B".

The following assignment applies:

| Motor data set | Coding | |
|---|---|---|
| MDS0 | 0 | 0 |
| MDS1 | 0 | 1 |
| MDS2 | 1 | 0 |
| MDS3 | 1 | 1 |

### Actual motor data set

The display (MDS0...3) of which motor data set is currently active corresponds to the interface signal:
DB31, ... DBX93.3 to 93.4 "Active motor A, B".

### Operating mode

The display of the control type of a drive corresponds to the drive parameter: p1300[0…n] "Open-loop/closed-loop control mode".

Depending on "n", the following "values" are displayed:

| n | Display |
|---|---|
| 20 | Speed control (without encoder) |
| 21 | Speed control (with encoder) |
| 23 | Torque control (with encoder) |

### Position actual value measuring system 1/2

The actual position of the axis measured via measuring system 1/2. The position is displayed in the machine coordinate system (no work offsets or tool offsets taken into account).

Unit: mm, inches or degrees

### Heat sink temperature

The display of whether the heat sink temperature is OK corresponds to the interface signal: DB31, ... DBX94.1 "Heat sink temperature prewarning".

| Status | Meaning | Display |
|---|---|---|
| 1 | The drive signals a "Heat sink temperature prewarning" to the PLC. | Overtemper-ature |
| 0 | The drive module heat sink temperature pre-warning has not responded. | OK |

### Motor temperature

The display of whether the motor temperature is OK corresponds to the interface signal: DB31, ... DBX94.0 "Motor temperature prewarning".

| Status | Meaning | Display |
|---|---|---|
| 1 | The motor temperature has exceeded the warning threshold configured in the drive. | Overtemper-ature |
| 0 | The motor temperature is below the warning threshold. | OK |

## Ramp-up function completed

The status display of the drive as to whether the ramp-up has been completed corresponds to the interface signal:
DB31, ... DBX94.2 "Ramp-up completed".

| Status | Meaning | Display |
|---|---|---|
| 1 | The actual speed value has reached the speed tolerance band specified via p2164 after a new speed setpoint specification and has not left the band for the duration of p2166. | Yes |
| 0 | The ramp-up procedure is still active after the speed setpoint has been changed. | No |

## Torque lower than threshold setting

The status display of the drive as to whether the threshold torque has been undershot corresponds to the interface signal:
DB31, ... DBX94.3 "$|M_d| < M_{dx}$".

| Status | Meaning | Display |
|---|---|---|
| 1 | The current torque utilization lies below the torque utilization threshold set in p2194. The drive signals the PLC that the torque setpoint $|M_d|$ does not exceed the threshold torque $M_{dx}$. | Yes |
| 0 | The torque setpoint $|M_d|$ is larger than the threshold torque $M_{dx}$. This signal can be used to determine whether the motor is overloaded. | No |

## Speed lower than minimum setting

The status display of the drive as to whether the minimum speed has been undershot corresponds to the interface signal:
DB31, ... DBX94.4 "$|n_{act}| < n_{min}$".

| Status | Meaning | Display |
|---|---|---|
| 1 | The actual speed value $|n_{act}|$ is less than the threshold minimum speed $n_{min}$ set in p2161. | Yes |
| 0 | The actual speed value is greater than the threshold minimum speed. | No |

## Speed lower than threshold setting

The status display of the drive as to whether the threshold speed has been undershot corresponds to the interface signal:
DB31, ... DBX94.5 "$|n_{act}| < n_x$".

| Status | Meaning | Display |
|---|---|---|
| 1 | The actual speed value $|n_{act}|$ is less than the threshold speed $n_x$ set in p2155. | Yes |
| 0 | The actual speed value is greater than the set threshold speed. | No |

### Actual speed = set speed

The status display of the drive as to whether the actual speed value matches the speed setpoint corresponds to the interface signal:
DB31, ... DBX94.6 "$|n_{act}| < n_{set}$".

| Status | Meaning | Display |
|---|---|---|
| 1 | The speed deviation between the setpoint and actual value is within the tolerance specified in p2163. | Yes |
| 0 | The speed deviation between setpoint and actual value is outside the specified tolerance. | No |

### Diagnostics for alarms

This information is also provided as a diagnostic tool for diagnosing the causes of alarms such as:

- Drive fault:
  ⇒ **SC1 drive alarm message** is set.
  ⇒ Check **specified drive data set**, **specified motor data set**, **DC-link voltage**.
- Alarm 25040 "Standstill monitoring",
  Alarm 25050 "Contour monitoring",
  Alarm 25060 "Speed setpoint limitation",
  Alarm 25080 "Positioning monitoring"
  ⇒ the drive enable may be missing (**PLC pulse enable** not present); this results in the display **Pulses enabled** = No.
- **Motor temperature** = exceeded
  ⇒ check the current motor temperature.

For details on the behavior of the NC control in response to individual alarms, and remedial action, please refer to:

### References

Diagnostics Manual

## 15.3 System utilization

For the NC areas, you have the option of displaying the system resources (utilization display) currently being used:

- Runtime for the position controller, interpolator and preprocessing
- Time required for synchronized actions
- Position controller and interpolator NC load
- Interpolator buffer level

**Note**

The utilization evaluation of the synchronized actions is only displayed when the value in MD11510 $MN_IPO_MAX_LOAD ≠ 0.

### Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key.
   Further horizontal softkeys are displayed.




3. Press the "System utilization" softkey.
   The "System Utilization" window opens.
   You can track the dynamic utilization display.




4. Press the "Stop" softkey to stop the display update.




5. Press the "Start" softkey to refresh the values.

## 15.4 Creating screenshots

You can create screenshots of the current user interface.

Each screenshot is saved as a file and stored in the following folder:

/**user**/sinumerik/hmi/log/screenshot

### Procedure

Ctrl + P Press the <Ctrl+P> key combination.
A screenshot of the current user interface is created in .png format.
The file names assigned by the system are in ascending order from "SCR_SAVE_0001.png" to "SCR_SAVE_9999.png". You can create up to 9,999 screenshots.

### Copy file




1. Select the "Start-up" operating area.




2. Press the "System data" softkey.
3. Open the folder specified above, and select the required screenshots.




4. Press the "Copy" softkey.

   - OR -




   Press the "Cut" softkey.




5. Open the required archive directory, e.g. on a USB flash drive and press the "Paste" softkey.

**Note**

You can also copy the screenshots using "WinSCP" to a Windows PC.

**Note**

If you wish to view the screenshots, then you can open the files in SINUMERIK Operate. On a Windows PC, you can open the data using a graphic program, e.g. "Office Picture Manager".

## 15.5 Machine identity

Important information about the machine that was previously only available on paper can now be stored electronically, e.g. for an end-user certificate. When the equipment is serviced, these entries can be viewed by means of remote diagnostics or can be transmitted for the purpose of registering the machine on the Siemens Service Platform Industry.

You can store the following data for each machine:

- Machine type
- Machine number
- Manufacturer information
- Dealer information
- End user information
- Component list of the control technology that has been installed
- Logbook entries for changes to the machine

You save the machine information and address information in the "identSNAPSHOT" window. You can either enter the data manually or import it from templates.

### Machine information

| Entries | Meaning |
|---|---|
| Machine No. | The unique machine number (serial number of the CompactFlash card) is displayed in the header. This number serves as information only and cannot be changed. |
| Machine name | The manufacturer assigns a unique machine number in this field which is stored in machine data MD17400 $MN_OEM_GLOBAL_INFO.<br>Note: The machine name is a mandatory field. No data can be stored until this field is completed. |
| Machine type | Machine type. |

### Address information

Under Addresses you can enter the following data for the manufacturer/manufacturer's regional office, dealer and end customer:

- Customer number
- Manufacturer name and, if required, Branch name / Dealer name / End user name
- Street, Postal code/ZIP code, Location, Country (can be selected from a selection list), Region/State
- Address of contact: Name, Phone, Fax, E-mail, URL

## 15.5.1 Entering machine-specific information

### Preconditions

You must have the following access rights to be able to enter or change machine-specific information.




Access level 1 (manufacturer)




Access level 2 (service)




Access level 3 (user)

### Procedure




1. Select the "Diagnostics" operating area.




2. Press the "Version" softkey.
   It takes some time to call the version display. While the version data is being determined a progress message box and the appropriate text are displayed in the dialog line.




3. Press the "Logbook" softkey.




4. Press the "Change" softkey.
   The "IdentSNAPSHOT" window opens.
5. Enter the machine name and the machine type.
   - or -
   Import a template: Import template (Page 277)













6. Select which address information you wish to enter. To do this, press one of the following softkeys:
   - "Manufacturer"
     If you press the "Manufacturer" softkey again, it changes to "Manufacturer branch".
   - "Manufacturer branch"
     Pressing the softkey once again changes it back to "Manufacturer".
   - "Dealer"
   - "End user"
7. Enter the address data in the respective text fields.
   - or -
   Import a template: Import template (Page 277)




Press the "OK" softkey to save the changes.

### See also

1. Documenting startup (Page 283)
2. Defining start-up (Page 283)

Making a logbook entry (Page 284)

## 15.5.2 Create a template

You use the templates to prepare the machine information and address information and transfer it to the controller. This means that it does not have to be manually entered.

Alternatively, you can use the "PridaNet identSNAPSHOT" program to export the machine and address information from the Service Platform Industry database and use it as a template. This ensures that the information is clear.

**Templates**

The directory HMI Data/Templates/Examples/Machine identity contains a selection of templates.

You will find the following templates on the controller:

- "dealer.xml" for dealer data
- "ma_types.xml" for machine types
- "oem.xml" for manufacturer data
- "oemsubs.xml" for manufacturer data of a regional office
- "user.xml" for end user data

## Procedure




1. Select the "Start-up" operating area.




2. Press the "System data" softkey.




3. Copy the files from the folder /HMI-data/templates/examples/machine identity to a data carrier.
4. Copy the files to your PC for processing.
5. After processing, copy the data to a data storage medium and transfer them to the controller.




6. Insert the copies into the folder HMI-data/Templates/Manufacturer/Machine identity or use the "Import data" softkey.

   The addresses or machine types are displayed in drop-down list boxes. When you select a list, the data stored for that address is automatically entered in the input fields.

## Entering dealer data

You enter the address data of any number of dealers in the file "dealer.xml". If you want 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>
```

You enter the address data of any number of manufacturers in the "oem.xml" file. If you want to enter additional manufacturers, copy the area <Manufacturer> to </Manufacturer> for each dealer.

You enter the address data of any number of manufacturer subsidiaries in the "oemsubs.xml" file. If you want to enter additional subsidiaries, copy the area <Manufacturer-Subsidiary> to </Manufacturer-Subsidiary> in each case.

You enter the address data of any number of users in the "user.xml" file. If you want to enter additional users, copy the area <User> to </User> in each case.

### Example of file "ma_types.xml"

You enter the available machine types in the file "ma_types.xml".

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

## 15.5.3 Import template

You read your templates directly into the controller. You select and import the files from a local drive, network drive or USB depending on the configuration. The data is then automatically copied into the following folder: /**oem**/sinumerik/hmi/template/identity or HMI-data/templates/manufacturer/machine identity.

## Templates

You have the capability of importing files with any name. The content of the files is evaluated. The new file name is derived from the contained information.

| File | Information |
|---|---|
| ma_types.xml | Machine types |
| dealer.xml | Dealer's data |
| oem.xml | Manufacturer's data |
| oemsubs.xml | Manufacturer's data of a regional office |
| user.xml | End customer data |

## Procedure




1. Select the "Diagnostics" operating area.




2. Press the "Version" softkey.

   It takes some time to call the version display. The progress of the data determination is displayed in the dialog line.







3 Press the "Logbook" and "Change" softkeys.




4. Press the "Import data" softkey.

5. Select the appropriate drive and the required file.




6. Press the "OK" softkey.

   The file is automatically copied to the correct location in the system.

   If the template only contains one data set, the data is transferred directly to the entry fields and saved.




7. If the entry fields already contain data, use the "Yes" softkey to confirm overwriting of this information.

8. If the template contains several data sets, select one data set from the list.

   The relevant data is transferred to the entry fields.




9. Press the "OK" softkey.

   The selection or the manual change is confirmed and saved.

## See also

Create a template (Page 275)

## 15.5.4 Save information

All the machine-specific information of the control is combined in a configuration via the user interface. You then have the option of saving the machine-specific information on the drives that have been set up.

### Procedure




1. Select the "Diagnostics" operating area.




2. Press the "Version" softkey.
   It takes some time to call the version display. While the version data is being determined a progress message box and the appropriate text are displayed in the dialog line.




3. Press the "Save" softkey.
   The "Save version information: Select Archive" window opens. The following storage locations are offered depending on the configuration:
   - Local drive
   - Network drives
   - USB
   - Version data (archive: Data tree in the "HMI data" directory)




4. Then press the "New directory" softkey if you wish to create your own directory.




5. Press the "OK" softkey. The directory is created.




6. Press the "OK" softkey again to confirm the storage location.
   The "Save version information: Name" window opens.
7. Specify the desired settings.
   - "Name:" input field
     the file name is pre-assigned with <Machine name/no.>+<CF-card number>. "_config.xml" or "_version.txt" is automatically attached to the file names.
   - "Comment:" input field
     You can enter a comment that is stored with the configuration data.
   - Version data (.TXT)
     Activate the checkbox if you wish to output the pure version data in the text format.
   - Configuration data (.XML)
     Activate the checkbox if you wish to output the configuration data in the XML format.
     The configuration file contains the data you entered under Machine identity, the license requirements, the version information and the logbook entries.




8. Press the "OK" softkey to start the data transfer.

## 15.5.5 Adding hardware components

Most of the hardware components supplied by Siemens are already electronically acquired. You can add further hardware components.

You can view and supplement the hardware components in the version screen at the user interface.

All of the hardware components are listed in the version display and in the configuration data.

### Precondition

You require access level 2 (service) to add more hardware components.

### Entering additional components

If new hardware components are added, they must be assigned to the appropriate categories. The "Select component" drop-down list box offers the following categories in the "Add HW Comp." window:

**Category**

- NCU/PLC
- Operator Panel
- PLC-peripheral devices
- Drive/Motor
- Cable
- Accessories/Miscellaneous

You record the following data in the "Enter Additional Components (Configuration Data)" window.

| Entries | Meaning |
|---|---|
| Name | Hardware designation |
| Version | Version of the program |
| Order No.[MLFB] | Order number |
| Serial number | Serial number |
| Number | Number of components |

## Procedure




1. Select the "Diagnostics" operating area.




2. Press the "Version" softkey.

   It takes some time to call the version display. While the version data is being determined a progress message box and the appropriate text are displayed in the dialog line.




3. Select the "Hardware" area and press the "Details" softkey. The "Version Data / Hardware" window opens.

   The "Add HW comp." softkey appears.




4. Press the "Add HW comp." softkey.

   The "Select component type" drop-down list box opens.

5. Select a category under which you would like to store the hardware information.

   The "Enter Additional Hardware Components" input window opens.

6. Enter additional hardware components via the keyboard.

   - OR -

   If you are using a CSV file (CSV = Comma Separated Values), for example a Service Platform Industry parts list, you can also read in the data.

   Select the storage location where the CSV file is located.
   Select the file and press "OK".




   The data is transferred from the CSV file into the table. It is read-in according to the CSV format of the Service Platform Industry parts list. For each component, quantity, MLFB, name and serial number are entered. The version is not saved and is not transferred.

   Example:

   QUANTITY;MLFB;Z_OPTION;SERIAL

   12;6AV7812-0BB11-2AC0;;

   1;6FC52030AB110AA2;;T-0815




7. Press the "OK" softkey.

   The data is written to the "versions.xml" configuration file and is therefore electronically acquired.

   All hardware components are displayed in the "Version Data / Hardware" window.

   The manually entered components are assigned a "+", e.g. Cable+.

### 15.5.6 Configuration data

The machine-specific information is stored in the configuration data. This file is stored in XML format and forms thee basis for further processing on the Service Platform Industry.

- The data can be read by remote diagnostics.
- The data can also be transferred directly to the Service Platform Industry.
  Further information at the address (https://www.siemens.com/sinumerik/register).

#### Configuration data

The configuration data includes:

- Machine-specific data that is stored in the "Machine identity" dialog box
- Hardware/software versions that are stored in the "Versions" dialog box
- Options requiring licenses, which are stored in the "Licensing" dialog box
- Logbook whose entries are stored in the "Logbook" dialog box

#### Storage path

Depending on how the drives have been configured, the configuration data can be stored in the available directories.

#### See also

Machine identity (Page 273)

Adding hardware components (Page 280)

Licensing (Page 75)

Logbook (Page 282)

## 15.6 Logbook

The logbook provides an electronic machine history.

The time and date of commissioning is recorded in the logbook, and if the machine is serviced, this can also be logged electronically. This helps to optimize the service.

Stored entries cannot be changed or deleted.

## 15.6.1 1. Documenting startup

### Precondition

As a minimum, the machine name/No., the customer No. and the country where the manufacturer is located must be entered into the machine identity.

### Procedure




1. The machine-specific information data are entered.
2. Start-up (commission) the machine.




3. Press the "1st startup completed" softkey.
   The "1.SETUP" entry is displayed in the "Machine logbook" window together with the time and date.

**Note**

**2. Perform commissioning**

On completion of first commissioning, you will be prompted to perform the 2nd commissioning.

If the 2nd commissioning is not performed, then you will be prompted at regular intervals to complete the 2nd commissioning.

### See also

Entering machine-specific information (Page 274)

## 15.6.2 2. Defining start-up

### Precondition

As a minimum, the country of the end customer must be entered into the machine identity.

### Procedure




The machine-specific information data are entered.

2. Start-up (commission) the machine.




3. Press the "2nd startup completed" softkey.
   The "2.SETUP" entry is displayed in the "Machine logbook" window together with the time and date.

**Note**

**Registering the machine identity on the Service Platform Industry.**

After completion of the 2nd start-up, you are prompted to send the machine identity to the Service Platform Industry database via the Internet.

## See also

Entering machine-specific information (Page 274)

## 15.6.3 Making a logbook entry

Using the "New logbook entry" window to make a new entry into the logbook.

Enter your name, company and department and a brief description of the measure taken or a description of the fault.

**Note**

**Setting line breaks**

If you wish to make line breaks in the "Fault diagnostics/measure" field, use the key combination <ALT> + <INPUT>.

The date and entry number are automatically added.

**Sorting the entries**

The logbook entries are displayed numbered in the "machine logbook" window.

More recent entries are always added at the top in the display.

## Procedure

1. The logbook is opened.
2. 

Press the "New entry" softkey.
The "New logbook entry" window opens.
3. 

Enter the required data and press the "OK" softkey.
You return to the "Machine logbook" window and the entry is displayed below the machine identity data.

**Note**

**Deleting logbook entries**

Up to the completion of the 2nd commissioning, you have the option to delete the logbook entries up to the time of the first commissioning using the "Clear" softkey.

### Searching for a logbook entry

You have the option for searching for specific entries using the search function.

1. The "Machine logbook" window is opened.
2. Press the "Find" softkey.




3. Enter the desired term in the search form. You can make a search according to date/time, company name/department or according to fault diagnostics/measure.

   The cursor is positioned on the first entry that corresponds to the search term.




4. Press the "Continue search" softkey if the entry found is not the one that you are looking for.

### Additional search option




Press the "Go to Beginning" softkey to start the search at the latest entry.




Press the "Go to End" softkey to start the search at the oldest entry.

## 15.7 Action log

### 15.7.1 Setting the action log

In the "Action Log Settings" window you specify whether the operating sequences are to be logged.

Here, you select which control actions should be logged, so that operator actions and sequences can be tracked and understood at a later date.

When the action log is active, you can display recorded traces in the system data using the following files:

- "actual_actionlog.com", actual action log report
- "actual_crashlog.com", crashlog backup

### Data that can be logged

| | |
|---|---|
| Logging on | Logging is activated or deactivated. |
| Alarm status change | Incoming and outgoing alarms are logged. |
| Keyboard actions | All actions on the operator panel front and on an external keyboard are logged. |

| | |
|---|---|
| Channel status change | NC/PLC states are logged via the information channel status. If this information can be obtained in a timely manner, in many cases these states can be used to verify the operation of the machine control panel. |
| Window switchover | The form and dialog names (assigned by the programmer) are logged each time a window is opened or closed. |
| Writing NCK/PLC data | The writing of NCK and PLC variables is logged. |
| File access | Copying to the NC is logged. |
| Function calls in the NCK (PI service) | Some program calls, for example, ASUB, are logged. |
| Curr. program status | Additional information for particular events is activated. For important alarms that require NC stop, NC start, or NC reset, actual values and the current block are recorded for example.<br>The program ascertains which channel and which spindle are used for the additional information. |
| Write interval file | Settings for updating files:<br>• "automatic": The action log saves the information to an internal buffer. If the buffer is full the entries are written to the CF card. The entries may be lost when the controller is switched off.<br>• "after every event": All entries are backed up directly, this prevents data loss due to a power failure or similar.<br>Notice: As CF cards only permit a limited number of write accesses, this setting is not recommended for normal operation.<br>• "time-controlled": New entries are backed-up for a particular length of time. An additional input field appears in which you can specify a time in seconds. |
| Save log as file for alarm(s) | The alarm numbers for which a "crash log" is generated are specified. The alarms are entered separated by a comma. |

## Generating a crashlog file

"actual_crashlog.com" is a backup file of the actual log when a critical event occurs (for example, Emergency Stop alarm).

The file is then no longer stored in the ring buffer of the action log and can no longer be overwritten with new entries.

The entries in the crash log file are only overwritten when yet another critical events occurs.

At the beginning, the file has no entries and is only written to after the following events:

- Interface signal DB19.DBX0.6 "Save teleprinter log" changes from 0 to 1.
- The alarm entered in the "Save log as file for alarm(s)" entry field is output.

### Interface signal

| DB19 (PLC → HMI) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| DBB00 | | Save action log | | | | | | |

### Procedure




1. Select the "Start-up" operating area.










2. Press the "HMI", "Diagnostics", and "Action log" softkeys
   The "Action Log Settings" window opens.
3. Activate the "Logging on" checkbox to activate the action log.
4. Check the relevant checkbox if you want to acquire particular events in the action log.

### See also

Displaying the log file (Page 287)

## 15.7.2 Displaying the log file

Loggable data that was logged when the action log function was active is displayed in the "Action Log File" window.

### Log files

The actual data can be displayed using the following files:

- "actual_actionlog.com"
- "actual_crashlog.com"

### Procedure




1. Select the "Start-up" operating area.




2. Press the "System data" softkey.
   The data tree is displayed.
   You will find the saved files in the HMI data folder under /logs/action log.

3. Select the desired com file and press the "Open" softkey or double-click the file.




4. Press the "Display new" softkey to update the display of events in the log (actual_actionlog.com).

   The entries made since the log file was called up are displayed.

## 15.7.3 Searching in the log files

You can search for specific events in the log.

### Precondition

The relevant log file is open.

### Procedure




1. Press the "Search" softkey.

   A new vertical softkey bar appears. The "Search" window opens at the same time.

2. Enter the desired search term in the "Text" field.




3. Position the cursor in the "Direction" field and choose the search direction (forward, backward) with the <SELECT> key.




4. Press the "OK" softkey to start the search.

   If the text you are searching for is found, the corresponding line is high-lighted.




Press the "Continue search" softkey to continue the search, as necessary.

- OR -




Press the "Cancel" softkey when you want to cancel the search.

### Additional search option




1. Press the "Go to End" softkey to scroll to the end of a large log. You get to the oldest entry of the recording.




2. Press the "Go to Beginning" software to return quickly to the most recent log entry.

## 15.7.4 Storing a log

You can store the displayed log in the "HMI data/Logs/Alarm logs" directory. It is stored as a binary file as well as ASCII file. You can read an ASCII file with any editor.

The following logs are generated:

- action.com (binary file)
- action.log (ASCII file)
- crash.com (binary file)
- crash.log (ASCII file)

### Requirement

The relevant log file is open.

### Procedure




1. Press the "Save log" softkey.
   The log is stored in the "HMI data/Logs/Alarm logs" directory.

**Note**

The "Save log" softkey is only available for files that have not yet been saved.

## 15.7.5 Structure of a log file

### Content of the log file

The following data is recorded for every logged event:

| | | | |
|---|---|---|---|
| HMI version | | NC version | |
| System | Entry keyword | | Date/time |
| | Entry text | | |
| | DETAILS | | |

| Entry | | Description |
|---|---|---|
| HMI and NC version output | | |
| System: | | |
| | HMI | Keystrokes, window change |
| | NC | Write variables, PI services |
| | MSG | Alarm entries |
| | USR | Entries generated by the machine manufacturer for the user |
| | ERR | Action log error, e.g. log file not found |
| Entry keyword: | | |
| | HMI_START | Entry of an HMI boot |
| | HMI_EXIT | Entry of an HMI shutdown |

| Entry | | Description | |
|---|---|---|---|
| | PLC_CRASH | Entry of a PLC crash | |
| | KEY_PRESSED | Entry of a keystroke | |
| | KEY_HOLD | Entry of a key hold | |
| | KEY_RELEASED | Entry of a key released | |
| | ALARM_ON | Entry of an incoming alarm event | See below "Alarm display" |
| | ALARM_OFF | Entry of an alarm going event | |
| | ALARM_ACK | Entry of an alarm acknowledgement event | |
| | OPEN_WINDOW | Entry of a window opening | |
| | CHN_STATE_CHANGED | Entry of a channel status change | |
| | OPMODE_CHANGED | Entry of an operating mode change | |
| | TOOL_CHANGED | Entry of a tool change | |
| | OVERRIDE_CHANGED | Entry of override changes | |
| | DOM_CMD | Entry of a download to the NC | |
| | PI_CMD | Entry of a PI service | |
| | WRITE_VAR | Entry of a write to NCK/PLC variable | |
| | AREA_CHANGED | Entry of an area changeover | |
| | NC_CONNECTION | Entry of an coming/going NC connection | |
| | USER | User entry via the OEM interface | |
| | ACTIVATED | Action log was activated | |
| | DEACTIVATED | Action log was deactivated | |
| | INTERNAL | Internal action log entry | |
| Date/time | | Date and time of the event<br>For HMI_Start, PLC_Crash or date change since the last entry, the intermediate period between the entry keyword and date is designated with an "-". | |
| Entry text | | Event data is written in plain text. | |
| DETAILS | | If additional information regarding an entry is known, e.g. a recorded IPO trace. | |

## Example

**Note**

**DB19.DBW24 interface signal**

The DB19.DBW24 interface signal used in the example does not apply to the SINUMERIK 840D sl. It is DB1900.DBW4 for the SINUMERIK 828D.

```
NCK    WRITE_VAR                                    18.02.2009  09:40:12
       ncu1.local wrote variable: DB19.DBW24 = 0
HMI    KEY_PRESSED                                  18.02.2009  09:40:12
       Key up: "Return" (1000004/ 0)
HMI    HMI_START ----------------------------       18.02.2009  09:34:15
       HMI started.
```

### Alarm display

```
MSG     ALARM                                                Date/time
        No.       Deletion criterion   Text
        DETAILS
```

| Entry | | Description |
|---|---|---|
| MSG | | |
| ALARM | | Type of alarm: ALARM_ON, ALARM_OFF, ALARM_ACK |
| Date/time | | Data and time of the event |
| No. | | Alarm number |
| Deletion criterion: | | Information as to how the alarm was acknowledged |
| | AUTOMATIC | Automatic acknowledgement |
| | POWER ON | Acknowledgement using PowerOn |
| | RESET | Acknowledgement using NCK reset |
| | NC START | Acknowledgement using NC start |
| | NC RESET | Acknowledgement using NC reset |
| | ALARM CANCEL | Acknowledgement using Alarm Cancel key |
| | RECALL | Acknowledgement using the Recall key |
| | HMI | Acknowledgement using HMI |
| | PLC | Acknowledgement using PLC |
| Text | | Display of the English alarm text |
| DETAILS | | If additional information regarding an entry is known. |

### Example

```
MSG   ALARM_ON                                    20.02.2009   14:25:37
      8020   POWER ON   : Option 'activation of more than 1 channels' not
      set
      DETAILS:
      Mode: JOG Program: canceled Channel: interrupted
      Program-Level information:
      Level Program running:                      Invoc        Offset
      1     /_N_MPF0                              0            0
```

## 15.7.6 Advanced settings

You define the size of the log file of the action log in the "actlog.ini" configuration file.

### Procedure

1. You can copy a sample configuration file "actlog.ini" from the following directory: /siemens/sinumerik/hmi/template/cfg.
2. Save or create the file in the /oem/sinumerik/hmi/cfg or /user/sinumerik/hmi/cfg directory.
3. Open the file and after "; ActionLogSize=", enter the file size (in bytes).

### "actlog.ini" configuration file

```
Template for the configuration of Action Log
;
; To activate the settings remove the
; commentary ';' at the beginning of the line
[ActionLogSettings]
;To change the path of the internal action log file use the following setting
;ActionLogPath=/user/sinumerik/hmi/ac_log/action.com
;To change the size of the internal action log file use the following setting
;ActionLogSize=5000000
;To change the path of the internal crash log file use the following setting
;CrashLogPath=/user/sinumerik/hmi/ac_log/crash.com
```

## 15.8 HMI trace

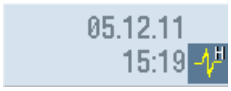
### Creating an HMI trace

The "HMI trace" function is available for extended diagnostics of the operating software. The prerequisite is at least access level 3 (user).

Procedure to create an HMI trace file:

1. An expert of the Service & Support Center provides a sltrc_keybitset.ini file. Copy the sltrc_keybitset.ini file to a USB memory medium and load the file on the controller or operator panel.
2. In the operating area, select "Start-up" → "HMI" → "Diagnostics >" → "HMI trace".
3. Select the INI file on the USB memory medium with the "Load configuration" softkey.

4. Start the trace by activating the checkbox or from any place on the user interface with the following key combination:
   <Alt> + <T> on the operator panel
   <Alt> + <Shift> + <T> on an external keyboard
   The following icon is displayed at the top right in the header to show that a trace is active:

5. On the user interface, repeat the operating sequence whose cause is to be clarified.
6. Stop the trace by deactivating the checkbox or from any place on the user interface with the following key combination:
   <Ctrl> + <T> on the operator panel
   <Ctrl> + <Shift> + <T> on an external keyboard
7. A sltrc.out output file is generated which you can copy to the storage medium with the "Export data" softkey.
8. Send the sltrc.out file for analysis back to the Service & Support Center.

**Note**

On completion of the analysis, it is recommended to delete all files under the following path: ../user/sinumerik/hmi/log/sltrc

**Note**

**"Extended" softkey**

The other options in this dialog support suitably qualified service engineers in the analysis of the system and evaluation of the HMI trace.

## 15.9 PROFIBUS diagnosis

Display the PROFIBUS status for diagnostic purposes during the configuration or when errors occur. This diagnostics window is only intended for information purposes. You cannot modify them.

**PROFIBUS connections**

- DP1 X126
- DP2 X136
- DP integrated

| Display | Meaning/information |
|---|---|
| **Status** | |
| Bus status | POWER_ON (0): State after the control has been switched on<br>OFFLINE (1): Basic initialization has been performed<br>STOP (2): Start in accordance with the hardware configuration (SDB)<br>CLEAR (3): PROFIBUS slaves have been parameterized and configured in accordance with the hardware configuration (SDB) and taken into the cyclic data exchange with zero output data<br>OPERATE (4): Cyclic data exchange with the PROFIBUS slaves running<br>ERROR (9): A fatal error had been detected (e.g. invalid or faulty SDB)<br>(<n>) corresponds to the OPI variable value for the bus state. |
| **Bus configuration** | |
| S7 Subnet ID | S7 Subnet ID of the PROFIBUS subnet |
| Baud rate | Transmission rate in Mbd |
| Cycle time | Configured bus cycle time in msec; also defines the position controller cycle at the same time |
| Sync. Component (TDX) | Configured time interval in msec for the cyclic data exchange within a PROFIBUS DP cycle |
| **PROFIBUS diagnostics/slaves** | |
| Slave address | Configured number of the DP slave |
| Assignment | Information whether the DP slave is assigned to the NC or to the PLC.<br>NC: e.g. one or more drives controlled by the NC<br>PLC: e.g. I/O peripherals or an axis controlled by the PLC<br>NC/PLC (for DP integrated) |
| Communication state | Information whether the DP slave is identified on the bus via the communication state.<br>Green: DP slave has been detected on the PROFIBUS DP and the data exchange is working error-free with the assigned component (NC and/or PLC)<br>Red: Incorrect response / no communication |
| Synchr. with NC | Information whether the DP slave is running synchronously to NC on the bus.<br>Green: DP slave runs synchronously to NC on the PROFIBUS DP, i.e. there is an isochronous data exchange<br>Red: Incorrect response / no communication<br>Gray: DP slave is not assigned to the NC, but to the PLC |
| Number of slots | Number of configured slots within the DP slave |

## References

Further information about the configuration of the properties of the network interface for PROFIBUS can be found in the following:

CNC Commissioning Manual: NCK, PLC, drive, SINAMICS S120

## Procedure




1. Select the "Diagnostics" operating area.



2. Press the "Menu forward" key and the "Bus TCP/IP" softkey.
   The "PROFIBUS Diagnostics" window is opened.







3. If several PROFIBUS connections have been configured, press the "Bus -" or "Bus +" softkey to select the desired configuration.




- OR -




Press the "Select bus" softkey and in the list that is displayed, select the required configuration.




Press the "OK" softkey.

## 15.9.1 Displaying details for DP slaves

Further information about the slots of a selected DP slave is displayed here.

| Display | Meaning/information |
|---|---|
| **Slave** | |
| Address | The DP slave selected in the PROFIBUS diagnostics / slaves pane |
| Communication state | Data regarding the NC and/or PLC assignment of the DP slave |
| In synchronism with NC | Data regarding NC synchronous operation of the DP slave<br>If a synchronous value is not available for the DP slave, then the display remains hidden. |
| **Slots** | |
| No. | Slot number within the DP slave |
| I/O addr. | I/O address in the I/O address space of the PLC, assigned to this slot<br>For NC axes, the setpoint and the actual value must always be configured on the same I/O address. |
| Type | Data as to whether the slot is input, output or diagnostic slot<br>If the slot is assigned to an NC axis, then the output is always denoted as setpoint value and the input always as actual value. |
| Length (bytes) | Length of the I/O area reserved for the slot in the STEP7 I/O address space |

| Display | Meaning/information |
|---|---|
| Communication state | Actual communication state of the slot<br>If a value is not available for the NC axes, then the display remains hidden.<br>Green: Slot used by the NC, communication active<br>Red: Slot used by the NC, communication currently not active<br>Gray: No NC axis |
| Machine axis | Display of the name defined in the machine data for this slot<br>If the slot is not assigned to any NC axis, then the display remains hidden. |
| Telegr. type (NC) | Displays the assigned telegram type<br>If a telegram type is not assigned in the NC machine data, then the display remains hidden. |

### Procedure




1. Select the "Diagnostics" operating area.



2. Press the "Menu forward" key and the "Bus TCP/IP" softkey.
   The "PROFIBUS Diagnostics" window is opened.




3. Select the PROFIBUS configuration for which you want to display details.




4. Press the "Details" softkey.
   The "PROFIBUS Diagnostics ... - Details" window is opened.

# 15.10 Drive system

## 15.10.1 Displaying drive states

### Status display

The "Drive System Diagnostics" window displays the status information about the drive units and the associated drive objects. The status symbols have the following meaning:

| Display icon | | Meaning |
|---|---|---|
| | Green | The drive object is in (cyclic) operation, without any detected problems. |
| | Yellow | The drive object has detected a less serious problem, i.e. a warning is pending, for example, or enables are missing. |
| | Red | This drive object has detected a serious problem, for example, an alarm is pending. |

| Display icon | | Meaning |
|---|---|---|
| ○ | Gray | The drive status could not be determined for this drive object. |
| # | Special characters | Error when reading the data. |

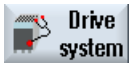
### Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key and the "Drive system" softkey.




3. The "Drive System Diagnostics" window opens.
   The name of the selected drive object is displayed in the window title.
   Press the "Drive object +" or "Drive object -" softkey.
   The next (+) or previous (-) drive object is selected.







   - OR -




   Press the "Select drive unit" softkey.
   The "Select Drive Unit" window opens.
4. Select the desired drive unit via the drop-down list box and press "OK" to confirm the selection.

### See also

Displaying details of the drive objects (Page 297)

## 15.10.2 Displaying details of the drive objects

### Drive system diagnostics - details

The following information is displayed for the drive objects:

- Status display (r0002)
  - The status display (parameter r0002) of the drive object is displayed.
- Missing enable signals (r0046)
  - The enable signals are diagnosed except for the drive objects that do not have enable signals (e.g. Control Unit). Missing enables are shown in a pane below the displays.

- Commissioning parameter filter
  - (p0009) – control unit
    The value of the "Device commissioning parameter filter" control unit parameter (p0009) is displayed.
  - (p0010) – further drive object
    The value of the "Drive object commissioning parameter filter" parameter (p0010) is displayed.
- Current fault
  The alarm number of the actual drive object fault is displayed or "No fault present", when no drive object fault is pending.
  You obtain an overview of the pending faults with the following data in an additional window:
  - Alarm number
  - Time: Date and time
  - Alarm text
- Current warning
  The number of the current drive object warning is displayed or "No warning present", when no drive object warning is pending.
  You obtain an overview of the pending alarms with the following data in an additional window:
  - Alarm number
  - Time: Date and time
  - Alarm text

## Meaning of the symbols

| Display icon | | Meaning |
|---|---|---|
| | Green | The relevant value of the drive object signals no problem detected. |
| | Yellow | The relevant value of the drive object signals a less serious problem, i.e. a warning is pending, for example, or enables are missing. |
| | Red | The relevant value of the drive object signals a a serious problem, for example, an alarm is pending. |
| | Gray | The drive status could not be determined for this drive object. |
| # | Special characters | Error when reading the data. |

## Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key and the "Drive system" softkey.




The "Drive System Diagnostics" window opens.

3. Select the drive object for which you want to display further details.




4. Press the "Details" softkey.

The "Drive System Diagnostics - Details " window opens.

The name of the drive object is displayed in the window header.




Press the "Faults" softkey.

The "Drive Unit Faults" window opens that shows an overview of the faults that have been output.




Press the "Alarms" softkey.

The "Drive Unit Alarms" window opens that shows an overview of the alarms that have been output.




You can display the pending SI messages if Safety Integrated is integrated in your controller.

# 15.11 Remote diagnostics

## 15.11.1 Adapting remote diagnostics

Depending on the operating mode, the initiative for remote diagnostics comes from the service provider or from the controller.

If the initiative comes from the controller, the "Request remote diagnostics" softkey is set-up.

The display of the "Request remote diagnostics" softkey is defined in the "tcu.ini" file under the [PINGSERVICE] section.

- Variable PingServiceMode
  0 = disable
  1 = enable

In additional variables of this section, the data for the request telegram for service should be set.

- Variable PingServerIP
  IP address of the service provider to which the request telegram should be sent, e.g. PingServerIP = 128.45.234.19
- Variable PingTransmissionData
  A text that should be sent out as data content in the request telegram,
  e.g. PingTransmissionData = help at machine 35
- Variable PingTransmissionPeriod
  The length in minutes that the telegram should be sent,
  e.g. PingTransmissionPeriod = 5

- Variable PingTransmissionInterval
  Duration in seconds after which send is repeated,
  e.g. PingTransmissionInterval = 15
- Variable PingServerPor
  Port number of the Ping service with the manufacturer,
  e.g. PingServerPort=6201

## 15.11.2 Remote diagnostics via Teleservice adapter IE at X127

The following settings must be made for remote diagnostics via modem using a Teleservice adapter IE (TS Adapter) that is connected at X127.

### Settings in the TS Adapter required

- IP address of the adapter: 192.168.215.30, subnet mask 255.255.255.224
- IP address of the caller: 192.168.215.29
  These two addresses are also stored as standard in the "tcu.ini" file. If other addresses are used, then appropriately change the entries "X127ModemIP" and "X127RemoteAccessIP" in the "tcu.ini" file.
- Enable the ports for the services for S7 communication, http, HTTPS, SSH as well as VNC (port 5900) in the firewall.

### Connecting the modem for remote diagnostics

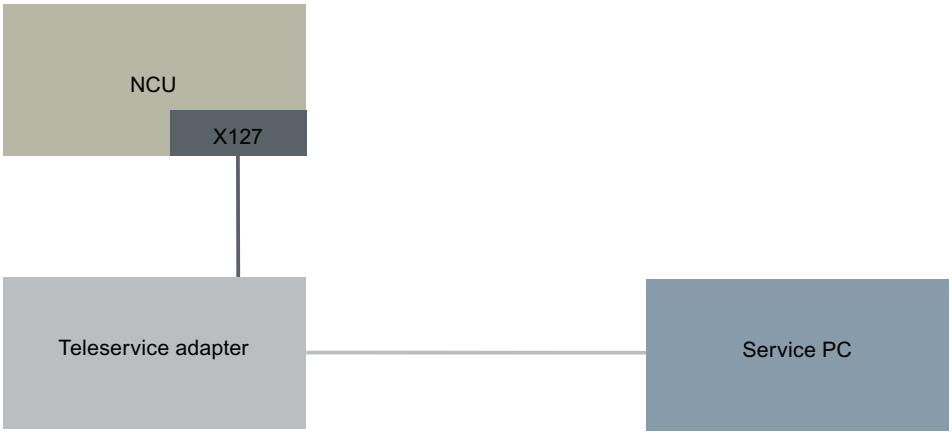



Figure 15-1 Basic system with SINUMERIK Operate on NCU (and operating station with TCU)

With the previously mentioned settings and the default values, in this configuration, remote diagnostics via TS Adapter is possible without any additional changes having to be made.

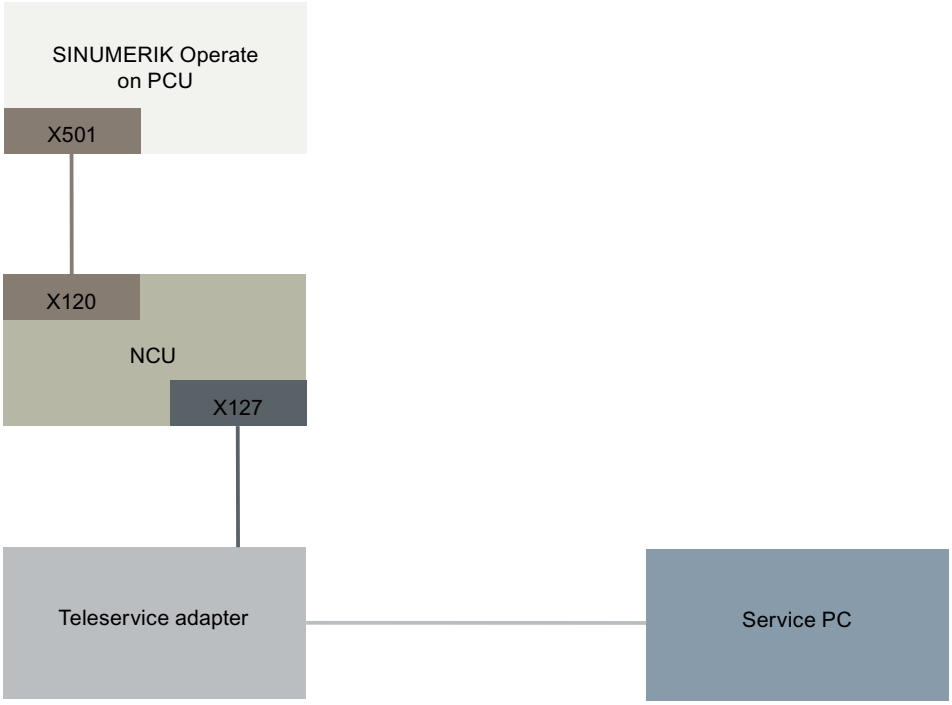



Figure 15-2 Basic system with SINUMERIK Operate on PCU

With the previously mentioned settings and the default values, in this configuration, remote diagnostics via TS Adapter is possible without any additional changes having to be made. When operating a SINUMERIK Operate on a PCU, the SINUMERIK Operate in the NCU must be switched off.

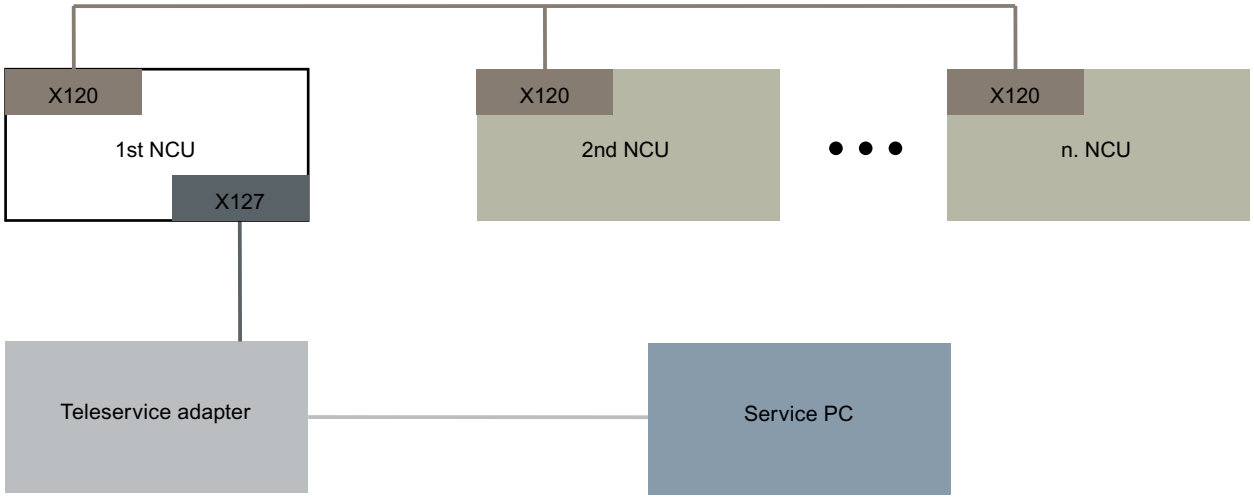



Figure 15-3 System with several NCUs and a fixed modem

The "Access MyMachine /P2P" option only has to be set for the NCU to which the TS Adapter is connected at the X127 interface. Modem access can only be permitted by making the appropriate entries at the SINUMERIK Operate in this NCU and the "Request remote control" function executed. If modem access is permitted, then the service PC implicitly has access to all other NCUs and their operating software. On the other hand, the operating screens to control

the remote access can be used and set at each station (SINUMERIK Operate in the NCU) individually for the station in conjunction with the relevant local PLC.

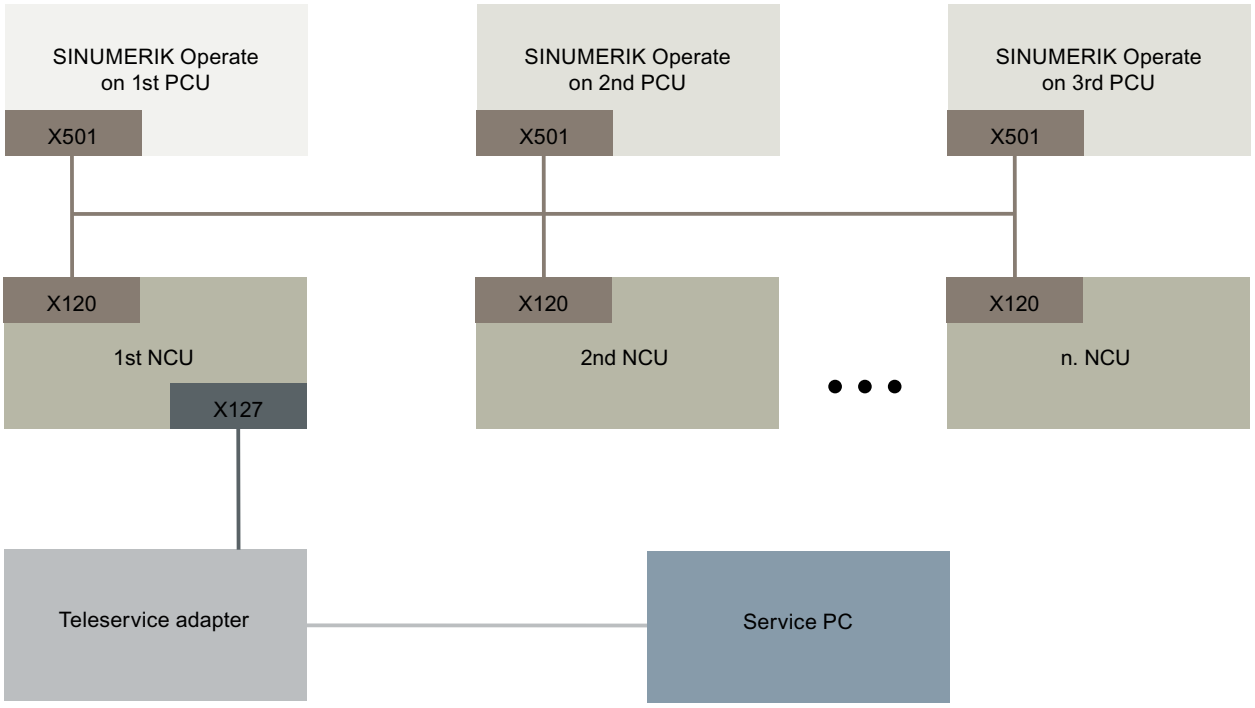



Figure 15-4 System with several NCUs, several PCUs and a fixed modem

The "Access MyMachine /P2P" option only has to be set for the NCU to which the TS Adapter is connected at the X127 interface. Modem access can only be permitted by making the appropriate entries at the SINUMERIK Operate on the PCU that is assigned to this NCU and the "Request remote control" function executed. If modem access is permitted, then the service PC implicitly has access via the plant or system network to all other NCUs and PCUs and their operating software. The operating screens to control the remote access can be individually used and set at each station (SINUMERIK Operate in the NCU) for the station in conjunction with the relevant local PLC.

## 15.11.3 PLC control for remote access

In the PLC user program, you can control external viewers using the following settings.

| | | | |
|---|---|---|---|
| DB19.DBX1.0 | PLCExtViewerReject | | |
| Value | = 0 | = 1 | |
| Meaning | Permit remote control | No remote control | |
| DB19.DBX1.1 | PLCExtViewerMode | | |
| Value | = 0 | = 1 | ignored |

| | | | |
|---|---|---|---|
| Meaning | Operating rights for remote control | Only monitoring for remote control | - |
| Result | Operating rights for remote control | Only monitoring for remote control | No remote control |

## 15.12 Trace

The trace provides you with an oscilloscope function that allows the following data (signals) to be graphically displayed:

- General NC data
- PLC data

### Trace session

The function serves for troubleshooting and resolving faults as well as to analyze the machine and process performance. In a session, data (signals) are recorded briefly before and after an event.

### Graphic display

The recorded data is optically displayed in the form of curves, the so-called characteristics graphs.

You have the option of individually setting the recording type.

### Saving and downloading data

You can store the recorded data in an XML file. The saved file can then be re-loaded so that it can be evaluated at a later time.

### Recording fundamentals: Signals (data)

Signals (data) specify what is to be recorded:

- NC variables (axis data, drive data, channel data, etc.)
- PLC data

Cyclic events

 Interpolation clock cycle (IPO clock cycle)

PLC cycle OB1

 Position control clock cycle (servo clock cycle)

Non-cyclic events



### Starting a session

- Recording using a trigger
- Manually starting and stopping a trace

### Analyzing the result

The following functions are available to analyze the result:

- Scaling characteristic graphs
- Zooming characteristic graphs
- Defining a characteristic range using markers

## 15.12.1 General procedure

#### Sequence

- Create a trace session file as follows:
  - Select the variable and define its representation type.
  - Set the graphics window, and the save mode
  - Set the trigger property to define the start and end of recording
- Save the recording and its associated properties that have been set
- Analyze the result by making the settings for the display in order to view additional details (e.g. zoom in, scroll, select section, etc.)

### See also

Trace options (Page 315)
Saving the trace file (Page 305)
Variable for trace (Page 307)
Trace settings (Page 313)
Evaluate a trace (Page 316)

## 15.12.2 Trace session

### 15.12.2.1 Creating a session file

#### Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key and the “Trace" softkey.










3. Press the "Load trace" and "New" softkeys.
   The "New Session" window opens in which the "Session" name is recommended. The file format is automatically specified as ".xml".

4. Supplement the file name or enter a new file name.




5. Select a variable type and press the "OK" softkey.
   You return to the "Select variables for trace:....” window.
   The name of the session that has been newly created is displayed in the header.

#### See also

Variables filter/search (Page 307)

### 15.12.2.2 Saving the trace file

There are two options of saving a trace session:

- Variables and settings: Only the variables and the settings are saved.
- Variables, settings and recorded values: The variables, settings and the recorded values are saved.

**Note**

**Save option**

If you have activated the "Save file in both formats" checkbox at "Trace" → "Options (Page 315)", the session file is saved as a csv file in addition to xml format.

## Procedure




1. You have edited an active trace session.




2. Press the "Store trace" softkey.
   The "Save Trace: Session ... " window opens and offers the following storage options.




3. Activate the corresponding checkbox and press the "OK" softkey.
   The "Store Trace" window opens and specifies the file name in the "Name of trace file:" entry field. The file format is automatically specified as ".xml" and cannot be changed.




4. Press the "OK" softkey to save the session file.
   If the session file already exists, then you obtain a corresponding confirmation prompt.
5. Press the "OK" softkey if you wish to overwrite the existing session file.
   - OR -




   Press the "Save as" softkey if you do not wish to overwrite the existing session file.
   A new name is suggested but you can specify a name yourself.
   - OR -




   Press the "Change directory" softkey if you wish to store the session file in a directory other than in the standard "Trace" directory.
   The data tree opens and you can select between the configured drives.




   Press the "New directory" softkey if you wish to create a new directory.
   The "New Directory" window opens in which you can assign a directory name.




   Press the "OK" softkey.
   The "Store Trace" window opens in which the names of the target directory and the session file are displayed.
   You can change the name of the session file.




6. To save the session file, press the "OK" softkey one more time.

   - OR -

   If the session file with this name already exists, then you obtain a corresponding confirmation prompt.

### 15.12.2.3 Load trace session file

You have the option of loading already saved session data, to restart them or to display the trace that has been recorded.

| Settings | Meaning |
|---|---|
| Variables, settings | You can restart a trace session that has already been created. |
| Variables, settings and recorded values | You can view the result of a trace session, change the variable and if required, restart. |

### Precondition

A session file of recorded traces is available.

### Procedure




1. You have selected the "Trace" function.




2. Press the "Load trace" softkey.
   The "Load Session File: Please Select File" window opens.




3. Select the desired session file from the appropriate directory, e.g. "Trace", and press the "Display trace" softkey.
   The "Load trace: session..." is opened.
4. Select the checkbox "Variables, settings", if you wish to immediately start the trace session.




5. Press the "OK" softkey and the "Start trace" softkey to restart the trace.




- OR -

Select the checkbox "Variables, settings and recorded values", if you wish to display the trace session.




Press the "OK" softkey.

The data of the selected session are loaded displayed in the "Trace" window.

## 15.12.3 Variable for trace

### 15.12.3.1 Variables filter/search

### Searching for PLC/NC/servo variables

To limit the selection of variables for PLC/NC/servo, set a filter and/or enter a start value for the search:

| Filter: | Display of the following variables: |
|---|---|
| System variable | System variables |
| NC | NC variables |
| Axis | Axis variables |
| Channel | Channel variables |
| Mode groups | Variables of the mode groups in OPI notation |
| Servo | Servo trace variables |
| Global user data | GUD such as: SGUD, MGUD, UGUD, GUD4, ... |

| Filter: | Display of the following variables: |
|---|---|
| PLC | PLC variables |
| PLC alarms | Manufacturer-specific PLC alarm signals |
| All (no filter) | All variables in alphabetical order |

## Inserting variables

The start value for "Filter/Search" of variables differs. For example, to insert the variable $R[0], enter the following start value:

- The start value is 0 if you filter according to "System variables".
- The start value is 1 if you filter according to "All (no filter)". In this case, all signals are displayed and shown in the OPI notation.

The GUD from the machine data is only displayed in the Search window for the variable selection when the associated definition file has been activated. Otherwise, the sought variables must be entered manually, e.g. GUD/SYG_RM[1]

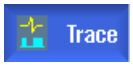
The following machine data is representative for all variable types (INT, BOOL, AXIS, CHAR, STRING): MD18660 $MN_MM_NUM_SYNACT_GUD_REAL

## Searching for drive variables

The restriction of the search is performed in this case via the selected drive object. In the Search window, you can restrict the selection by entering alphanumeric values, for example. Further restrictions of the selection are possible via the "All" or "Recommended" interconnections.

## Procedure

For a trace session, in the "Selected Variables for Trace: Session ... " window, set the variables whose signals are to be traced.




1. In the "Select variables for trace: Session..." window, a trace session is displayed.
2. Position the cursor in the "Variable" column in an empty line.

   - OR -

   Position the cursor in the "Variable" column on a variable that you wish to replace by another.




3. Press the "Insert variable" softkey.

   The "Select Variable" window opens.

   **Note:**

   When selecting a "Drive parameter", firstly select the drive object (DO) and then insert the desired parameter from the list of corresponding parameters.




4. Press the "Filter/search" softkey to start a specific search.

   In the "Filter" selection box, select a variable area.

   - OR -

Directly enter the variable into the "Search" entry field.

The cursor jumps to the term that is found.




Press the "Delete search" softkey if you wish to terminate the search.




5. Press the "Add" softkey to enter the required variable into a trace session.

   The "Variable Attributes" window opens and you can select the appropriate value (e.g. channel data, drive bus no.) from a combobox.

   - OR -




   Press the "Replace" softkey to replace an already selected version by another one.




6. Press the "OK" softkey to add or replace the variable.

   Each variable is automatically allocated a reference number, which is displayed in the upper section of the window. This number is opened automatically allocated. If a version is removed, then the following variables are re-numbered.

7. Now, select the desired settings for display, such as color, pen, qty., event. etc.

### Editing variables of a trace session

| Key combination | Effect |
|---|---|
| **Ctrl + X**<br>or right-click, shortcut menu "Cut" | Press the key combination or open the shortcut menu in order to cut out a variable. |
| **Ctrl + C**<br>or right-click, shortcut menu "Copy" | Press the key combination or open the shortcut menu in order to copy a variable. |
| **Ctrl + V**<br>or right-click, shortcut menu "Paste" | Press the key combination or open the shortcut menu in order to insert a variable. |

## 15.12.3.2 Selecting attributes of a variable

### Attributes

For a trace session, select the variables and assign the corresponding attributes:

| Column | Meaning |
|---|---|
| Variable | Address of the variables |

| Attributes | Meaning |
|---|---|
| Comment | • Name of the selected variables<br>• A general description for the variables. |
| Color | Color selection, the representation of the characteristic graphs.<br>When creating a variable, the next color from the available palette is automatically used. |

| Attributes | Meaning |
|---|---|
| Pen | Line type selection to display the characteristic graphs.<br>• No line<br>• Solid line<br>• Dashed line<br>• Dash-dot line<br>• Dash-dot-dot line<br>• Dotted line<br>• Stepped line XY<br>• Stepped line YX<br>• "* line" (line made up of * symbols)<br>• + line<br>• X line |
| Qty. | Defines whether the characteristic of the variables is displayed in the "Trace" window. If the checkbox is not selected, then no characteristic graphs are displayed. |
| Events | Event selection which triggers that the signal is traced: |
| | • Interpolation clock cycle (IPO clock cycle)<br>• Interpolation cycle (IPO2)<br>• Position control clock cycle (servo clock cycle)<br>• PLC cycle OB1<br>• Drive trace rate<br>• Start geometry axis/direction change<br>• Stop geometry axis<br>• Machine axis start / direction change<br>• Machine axis stop<br>• NC start (program runs)<br>• NC start (program ran)<br>• Start of the data recording |

| Attributes | Meaning |
|---|---|
| | • Block start, type 1 or block end (all program levels, without intermediate blocks)<br>• Block start, type 2 or block end (all program levels, with intermediate blocks)<br>• Block start, type 3 or block end (all main programs, without intermediate blocks)<br>• Block start or block end, block search (all program levels, without intermediate blocks)<br>• Geo axis start / direction change (2nd event)<br>• Geo axis stop (2nd event)<br>• Block start, type 2 (2nd event: all program levels, without intermediate blocks)<br>• Block end, type 2 (2nd event: all program levels, with intermediate blocks)<br>• Block start, block advance, type 1 (2nd event)<br>• OEM test event 1 (non-cyclic, block advance)<br>• OEM test event 2 (non-cyclic, main run) |
| | • Activate/deactivate synchronized action<br>• Synchronized action initiated (condition fulfilled)<br>• Signaled alarm<br>• Cancel softkey pressed<br>• Program level change<br>• Block end (Interpreter)<br>• Trace end (last event!)<br>• Start trace |
| | • WRTPR part program command<br>• WRTPR part program command (block search)<br>• Tool change<br>• Cutting edge change<br>• Tool change (block search)<br>• Cutting edge (block search)<br>• Start trigger initiated<br>• Stop trigger initiated<br>• Block end (block advance)<br>• Block end (2nd event: Block search)<br>• WRTPR part program command (block advance)<br>• Tool change (block advance)<br>• Cutting edge (block advance) |
| Event channel | This means that the channel can be defined in SINUMERIK in which a specific event occurs. For events which are not channel-specific, the field remains empty. |

| Attributes | Meaning |
|---|---|
| Bit mask | In the case of an integer variable, a bit mask can be specified in this field. For a bit mask, the particular signal value is AND'ed before being displayed. After the mask has been applied, all of the selected bits are moved to the right so that it looks as if the bit or the bits all start with bit zero.<br>This means, if all bits with the exception of bit 7 have been masked, then the integer number that would have been obtained would either have a value of 0 or 1, however, not 0 or 128.<br>If all of the bits with the exception of bit 7 and 0 have been masked, the resulting integer number would either have the value 0, 1, 2 or 3, however, not 0, 1, 128 or 129. |
| Decimal places | This setting is used to define how many places to the right of the decimal point are displayed at the axis identifiers. |
| Coord. axis | The coordinate axes are displayed to the left or right in the graphic window or there is no display. |
| Display Y | Value input or 0 |
| Scale factor | Defines the scale. |
| Units | Displays the measurement unit, e.g. mm/min. The system specifies this and it cannot be changed. |

### 15.12.3.3 Displaying details of a variable

The most important information and settings are displayed at a glance in the detail window:

- Variable address
- Comment with a description of the variables
- Events
- Channel, axis, access level, mode group, etc.
- Smaller graphic display with the event; when this event occurs the characteristic graphs are recorded as well as their attributes, such as color or line type.

## Procedure




1. You are in the "Select variables for trace: ..." window.




2. Position the cursor to the required variable, and press the "Details" softkey.

   A subscreen with the most important information as well as the graphic display is shown in the lower half of the window.




3. Press the "Details" softkey again to display the subscreen.

## 15.12.4 Trace settings

### 15.12.4.1 Trace settings (PLC, NC, servo)

In the "Trace Settings (PLC, NC, Servo)" window, you define the conditions for starting and stopping the trace session.

#### Condition for Start trace

| | |
|---|---|
| With "Start trace" softkey | The recording is started immediately after pressing the "Start trace" softkey. |
| If variable | The session is activated by a variable. The desired variable is selected via the "Insert variable" softkey or "Quick list" softkey. |
| • Condition and value | After the session has been activated, this defines at which value the trace recording is started.<br>The following values are offered: = =, ≠, >, >=, <, <=, ascending, descending, changed. |
| **Data acquisition [s]** | |
| Specifies where the recorded data is saved:<br>• With "On NCK", the recorded data is saved in the NC work memory until the recording is stopped. Only then is the session file saved on the CompactFlash card.<br>• With "CF card" or "On hard disk", the data is saved during the recording on the CompactFlash card or hard disk (e.g. of the PCU). As a consequence, more signals can be processed. | |

#### Condition for Stop trace

| | |
|---|---|
| With "Stop trace" softkey | The recording is stopped after pressing the "Stop trace" softkey. |
| If the time has expired | Defines the period which after expiry, the trace recording is exited. |
| If the memory is full | The trace session is terminated when the specified data volume (KB) is reached. |
| If variable | The recording is stopped by a variable. The desired variable is selected via the "Insert variable" or "Quick list" softkey. |
| • Condition and value | After the session has been activated, this defines at which value the trace recording is exited.<br>The following values are offered: = =, ≠, >, >=, <, <=, ascending, descending, changed. |
| • Delay time [s] | This defines how long the trace still runs after the session has been exited. |
| • Overwrite at memory limit (KB) | Defines from which memory limit the data is overwritten (ring buffer). |

### 15.12.4.2 Trace settings (drive)

In the "Trace Settings (Drive)" window, you define the conditions for starting and stopping the trace session:

## Condition for Start trace

<table>
<tr><td>With "Start trace" softkey</td><td colspan="3">The recording is started immediately after pressing the "Start trace" softkey.</td></tr>
<tr><td>If variable</td><td colspan="3">The recording is activated after pressing the "Start trace" softkey when the selected variable clearly satisfies the condition and the specified value.<br>The following conditions are offered:<br>• Positive edge<br>• Negative edge<br>• With bit mask<br>The "Bit mask" softkey is displayed when the "Bit mask" condition is selected in order to select the desired bits. Irrespective of the maximum value (hexadecimal) which is possible through the bit mask, a "Value" (decimal) can be entered at which the trace recording starts precisely.<br>**Example:**<br>The trace recording is started when the following bits are set: Bit 0 = 0; bit 1 = 1; bit 2 = 0; bit 3 = 1</td></tr>
<tr><td></td><td>**Bit mask = fH**</td><td>**Condition**</td><td>**Value (decimal)**</td></tr>
<tr><td></td><td>Variable: CU/ r722/3.3.1</td><td>With bit mask<br>☑ Bit 0<br>☑ Bit 1<br>☑ Bit 2<br>☑ Bit 3</td><td>10</td></tr>
<tr><td>Preprocessing time [s]:</td><td colspan="3">The input of a preprocessing time corresponds to the functional parameter p4722 "Trace trigger delay":<br>• A positive preprocessing time starts the recording **before** the trigger event.<br>• A negative preprocessing time starts the recording **after** the trigger event.<br>When entering a preprocessing time, the entire duration of the recording must be taken into account:<br>• If the preprocessing time is greater than the entire duration of the recording, the entry is rejected.<br>• If too large a preprocessing time is entered, the trigger event may not be recorded.</td></tr>
<tr><td colspan="4">**Data storage on CU**</td></tr>
<tr><td>Drive trace rate [µs]:</td><td colspan="3">Enter a factor with the "Sampling rates" softkey to set the effective sampling rate. With a higher sampling rate, the recording duration is also longer.</td></tr>
</table>

## Condition for Stop trace

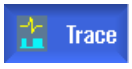
| | |
|---|---|
| If the time has expired | Defines the period which after expiry, the recording is exited. |
| If the memory is full | The recording is terminated when the specified data volume (KB) is reached. |

### 15.12.4.3 Trace options

You define the following properties of a trace session in the "Trace - Options" window.

- Setting the graphic window
  - Displaying main grid lines: You define whether a horizontal grid is displayed and in which color.
  - Displaying grid sub-lines: You define whether a vertical grid should be displayed and in which color.
  - Background: You select the background color of the graphic window.
- Defining storage options
  - Activate the "Confirm file overwrite" checkbox in order to receive a confirmation prompt each time a trace session is to be overwritten. If the checkbox is deactivated, the trace session is overwritten without a confirmation prompt.
  - Activate the "Save file in both formats" checkbox in order to also save the data as a csv file (Page 305) in addition to the xml session file. The file in csv format cannot be imported again.

#### Procedure




1. You are in the "Select Variables for Trace: ..." window and a trace session has been selected.




2. Press the "Options" softkey.
   The "Trace - Options" window opens.




3. Select the desired settings and press the "OK" softkey in order to accept the settings.

### 15.12.4.4 Starting the trace

In the "Trace" window, you obtain the curve with the data of the variables selected for a trace session.

The diagram comprises two Cartesian coordinates:

- Time axis
  The horizontal x axis supplies the time values.
- Value axis
  The vertical y axis displays the resolution fineness (amplitude).

**Note**

**Display of real values of PLC variables**

If the value format is displayed incorrectly in the trace, extend the variable specification in the "Select variable for trace" window with the parameter assignment :REAL, e.g. to DB21.DBD76:REAL. The real values of the variable will then be displayed in the trace.

**Recording the trace session**

- Manually starting and stopping a trace
- Starting and stopping recording using a trigger

## Procedure




1. You have listed the desired variables for a trace section.




2. Press the "Display trace" softkey.
   The "Trace" window is opened. The recording is loaded and the procedure is displayed in progress window.




3. Press the "Start trace" softkey if you wish to immediately start recording.

   - OR -

   The recording is triggered by the trigger condition.




4. If you wish to manually end the recording, press the "Stop trace" softkey.

   - OR -

   The recording is stopped by the trigger condition.

## 15.12.5 Evaluate a trace

### 15.12.5.1 Setting trace views

In order to permit a precise evaluation of the recorded signals, you have the option to edit the view as well as the display of the characteristic graphs:

- Selection/legend
  - Show legend: In addition to the graphic display, the data of the variable is displayed as comment, numbered and in color.
  - Make a selection: You define which variables are displayed in the graphic window.
- Scale
  - Specifying scaling factors for minimum and maximum X and Y value

- Zoom
  - Increasing or decreasing the section
- Cursors: The cursor display provides functions that allow one, two or both curves to be precisely measured.
  - Cursor A
  - Cursor B
  - Both cursors
  - Snap to curve: The cursor precisely moves along the curve
  - Point mode: Cursor jumps directly to various values.
  - High point mode: The cursor jumps from one peak value to the next.
  - Low point mode: Cursor jumps to the lowest values.

### Requirement

The "Trace" graphic window is open and the trace has been recorded.

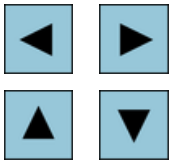
### Editing curves




You jump from one variable to the next using the <TAB> key.




You jump back using the <Ctrl> and <TAB> keys.



With the cursor keys, you set marks, move to individual values and can define a zoom range.

## 15.12.5.2 Selecting a variable

You have the option of displaying the selected variables as legend and to make a selection using a checkbox.

| Legend |
|---|
| Numbering the variables |
| Axis identifier |
| Comment of the variables |

### Displaying a legend and selecting a variable




1. A recording is opened in the "Trace" graphics window.




2. Press the "Legend / selection" softkey.
   The graphic window is halved. The variables are displayed as comment and with checkbox in the left-hand section of the window.
3. Activate or deactivate the required variable using the checkbox to display it as curve.



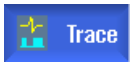

4. Press the softkey again to hide the legend area again.

## 15.12.5.3 Scaling the display

The scaling allows you to make a precise measurement of selected variables from the X time axis as well as from the Y value axis. You have the option of saving the larger or smaller display in a user-defined way

| Setting | Meaning | |
|---|---|---|
| X minimum or<br>X maximum | Highest or lowest value of the X time axis. The values are used for recording when the trace is started again. | |
| | Fixed scale, X axis | Values of the X time axis are kept. |
| | Strip chart recorder mode | The values of the X time axis are used while recording. |
| Y minimum or<br>Y maximum | Defining the highest or lowest value of the Y value axis (amplitude). | |
| Offset or<br>resolution | Defining the rms values (mean values) of the amplitudes.<br>The scale is changed by a factor of 10. | |

### Scaling the time and value axis




1. A recording is opened in the "Trace" graphics window.




2. Press the "Scale" softkey in order to scale the display.




The "Select X-Y Scaling" input window opens and the values of the X time axis are displayed.




3. Press the "X time axis" softkey again if you wish to hide the values.

- OR -




Press the "Y selected curves" softkey to display the values of the Y value axis.




Press the softkey again if you wish to hide the values.

- OR -

You have selected both axes.

Scale +

Scale -

4. Press the "Scale +" or "Scale -" softkey until the desired scaling is reached.

- OR -

Enter the scaling values directly into the input window and select the appropriate property.

Set scale

5. Press the "Scale" softkey to save the settings.

- OR -

Fit selected

Press the "Adapt selected" softkey to display the selected characteristic graphs in the full graphic window.

Fit one

Press the "Fit one" softkey to display the selected characteristic graphs in the full graphic window.

### 15.12.5.4 Zooming the display

Using the zoom function, you have the option of viewing selected sections of curves in detail. After you have defined an area or a section, you can increase or decrease the size of it (zoom in or zoom out).

## Procedure

Trace

1. A recording is opened in the "Trace" graphics window.

Zoom

2. Press the "Zoom" softkey if you wish to zoom-in or zoom-out on the characteristic graphs.

X - time axis

Y - value axis

3. The "X time axis" or "Y value axis" softkey is selected.

Fit separated

4. Press the "Adapt individually" or "Adapt all" softkey if you wish to display all selected characteristic graphs in a separate area positioned one above the other.

or

Fit all

- OR -

Fit overlaid

Press the "Adapt together" softkey if you wish to adapt all of the characteristic graphs to the graphic window.
As all of the curves fill the complete window area, they are displayed superimposed on one another.

- OR -

Fit X separated

Press the "Individually adapt X" softkey if you wish to normalize the X time axis only in the graphic window.

- OR -




Press the "Individually adapt Y" or "Adapt all" softkey if you only wish to scale the Y value axis in the graphic window.

or




- OR -




Press the "Adapt Y together" softkey if you wish to adapt the value axis of all characteristic graphs to the graphic window.
As all curves fill the complete window area, they are displayed superimposed.







5. Press the "Zoom +" or "Zoom -" softkey until the desired display size is reached.

**Zoom in/zoom out area**




Press the "Zoom area" softkey to define a specific section.

A frame is displayed.







Press the "Zoom +" or "Zoom -" softkey to change the frame step-by-step.

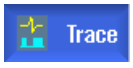



Press the "OK" softkey to display the zoom area.

### 15.12.5.5 Position the cursor

Using the cursor, you have the option of determining the precise values from a defined section. To do this, the cursor is positioned directly on the curve.

## Procedure




1. A recording is opened in the "Trace" graphics window.




2. Press the "Cursor A" softkey.
The reference number, color of the variables and the color of the coordinate crossing point are displayed on the softkey.




3. Position "(A)" at the position on the curve where you wish to start the measurement.
   The coordinates are specified in a tooltip and in the status line.




4. Press the "Cursor B" softkey.
   The reference number, color of the variables as well as the color of the coordinate crossing point are also displayed on this softkey.
   Position "(B)" at the position on the curve where you wish to end the measurement.







5. Press the "Both cursors" softkey to display the area between A and B.
   The coordinates of A and B as well as the difference "Δ" are displayed in the status line.




Press the active "Cursor A" and "Cursor B" softkeys again to deselect the position.
When re-selected, the positions are reset.







Press the "Crop to screen" softkey to display the area, defined by "Cursor A" and "Cursor B", in the complete graphic window.



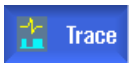

Press the "Snap to waveform" softkey to move the cursor along the curve step-by-step.

### 15.12.5.6 Acquiring measured values

You have the option of directly determining the following values of a curve:

- Peak-to-peak value
- Peak-to-valley value

### Procedure




A recording is opened in the "Trace" graphics window.




Press the "Point Mode" softkey to continuously move the cursor along the curve.




Press the "Peak Mode" softkey to position the cursor at the highest value.




Press the "Valley Mode" softkey to position the cursor at the lowest value.




If you press the "Peaks" softkey, the cursor only goes to the peak-peak values.




If you press the "Minimum" softkey, the cursor only goes to the peak-valley values.

# 15.13 PROFIBUS/PROFINET and AS-i bus diagnostics

## 15.13.1 PROFIBUS/PROFINET

With the PROFIBUS/PROFINET diagnostics, you obtain a quick overview from which, when required, you can call status and detailed information for external DP master systems and PROFINET IO systems.

### Precondition for detailed diagnostics

When using HMI PRO RT, the diagnostics can be extended by additional information (e.g. module and Order No.). This data involves offline data, which can be extracted from the associated Step7 project.

- You can create offline data using the HMIPRO CS program.
- You transfer data to the control using the HMIPRO program.



**Software option**

For faulted modules, in order that in addition to the status information, detailed information can also be displayed, you now require the following option: "Operating software SW HMI PRO sl RT".

In order that detailed information can be displayed for PROFINET-IO systems, the PLC block FB_SL_COM from the HMI PRO block library must be used. Use DB449 as the associated instance DB.

**Note**

Detailed information for DP master systems is also available without using FB_SL_COM.

### Diagnostics information from the DP master systems

You can configure the following versions:

1. Software option not available and HMIPRO offline data not available
2. Software option available and HMIPRO offline data not available
3. Software option available and HMIPRO offline data available

## 15.13.2 Displaying PROFIBUS/PROFINET diagnostics

### Diagnostics information from the DP master systems

From the user interface, you can display the fieldbus diagnostics known from HMIPRO. The module state is displayed using diagnostic symbols.

The following diagnostics information is displayed depending on the particular version that you have configured:

| **Information for all versions** | | |
|---|---|---|
| Status |  | Module has a fault |
| |  | Module has failed |
| |  | Module is OK |
| |  | Module is suppressed (this is only possible with HMI PRO RT) |
| PROFIBUS address | | PROFIBUS address data |

| **Additional information for the second version** | |
|---|---|
| PROFIBUS/PROFINET module or channel diagnostics | |
| Device diagnostics | For Siemens DP/DP coupler and diagnostics repeater |

| **Additional diagnostic information for the third version** | |
|---|---|
| Name | Module designation from the HMI offline data |
| Name | Module name from the HMI offline data |
| Order no. | Order number of the module from the HMI offline data |
| IP address | IP address of the module from HMI offline data (only for PROFINET) |
| Graphic display of the complete hardware configuration in a dedicated window | |

### Procedure




1. Select the "Diagnostics" operating area.



2. Press the "Menu forward" key twice and the "PB/PN diag." softkey.
   The "PROFIBUS/PROFINET Diagnostics" window opens.




3. Select the required DP master system.
   The complete diagnostics information is displayed.






4. Press the "Available devices" softkey to display the diagnostics data of the available modules.

   - OR -




   Press the "Suppressed devices" softkey to display the diagnostics data of the suppressed modules.

   - OR -




   Press the "Failed devices" softkey to display the diagnostics data of the failed modules.

   - OR -




   Press the "Faulty devices" softkey to display the diagnostics data of the faulty modules.

## 15.13.3 Configuring the AS-i bus

You obtain a fast overview of the existing AS-Interface network using the AS-Interface (AS-i) diagnostics. Permanent monitoring and an error search function provide you with the following information:

- Status flags acc. to the AS-i specification
- States of the AS-i slaves

### Precondition

- For AS-i diagnostics, it is absolutely necessary to use the FB_SL_COM PLC block from the HMI PRO block library.
- The configuration file "sltlprodiaghmi.ini" must be available.

### AS-i components

The following AS-i components are supported:

| | |
|---|---|
| CP142-2 | (Type ID 1) |
| CP343-2 | (Type ID 2) |
| DP/AS-i link | (Type ID 3) |
| DP/AS-i link 20E | (Type ID 4) |
| DP-AS-i link Advanced | (Type ID 6) |

### Diagnostic information

All of the AS-i components to be diagnosed must be parameterized in the "sltlprodiaghmi.ini" configuration file.

| Entry | Description |
|---|---|
| `[SKO]` | |
| `ComboCount=x` | Number of entries (x) in the selection list |

| Entry | | Description |
|---|---|---|
| [SKO_Index0] | | 1. Entry |
| | DiagType= 6 | Type ID of the AS-i component (refer above) |
| | TypeSection=DP-AS-i-Link-Advanced 1 | Designation of the AS-i component, which should be output in the selection list |
| | Logical_Address=43 | E address of the AS-i component from the hardware configuration |
| | RequestDB=449 | Instance DB of the function block "FB_SL_COM" being used |

## Example of the "sltlprodiaghmi.ini" with 5 components

```
[SK0]
ComboCount=5

[SK0_Index0]
DiagType= 1
TypeSection=CP142-2
Logical_Address=43
RequestDB=449

[SK0_Index1]
DiagType= 2
TypeSection=CP343-2
Logical_Address=75
RequestDB=449

[SK0_Index2]
DiagType= 3
TypeSection=DP-AS-i-Link
Logical_Address=75
RequestDB=449

[SK0_Index3]
DiagType= 4
TypeSection=DP-AS-i-Link-20E
Logical_Address=75
RequestDB=449
```

```
[SK0_Index4]
DiagType= 6
TypeSection=DP-AS-i-Link-Advanced
Logical_Address=75
RequestDB=449
```

### Procedure

1. Copy the "sltlprodiaghmi.ini" file from the /**siemens**/sinumerik/hmi/template/cfg directory.
2. Place the file in the directory: /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg.
   If the file is already available in the directory, then just supplement this by the corresponding entries.
3. Open the file and enter the corresponding data of the AS-i components.
4. Save and close the file.
   The diagnostics information is displayed on the user interface.

## 15.13.4 Displaying AS-i bus diagnostics

### Diagnostics information from AS-i components

Using the user interface, you have the option of displaying information about the networking of binary sensors and actuators (AS-i).



Software option

Additional error counter can be displayed for the component DP-AS-i Link Advanced if the option "Operating SW HMI PRO sl RT" is set.

### Diagnostic information

| Flags acc. to the AS-i specification | Description |
|---|---|
| Offline | Offline operating state |
| Internal | Internal operating state |
| EEPROM o.k. | Internal EEPROM okay |
| Auto-Addr.progr. enabled | Automatic address assignment enabled |
| I/O fault | Peripheral fault present |
| Master Power On - Start | Starting after Master Power On |
| Offline ready | Offline phase completed |
| AS-i cable voltage too low | AS-i cable voltage too low |
| CP in normal operation | CP in normal operation |

| Flags acc. to the AS-i specification | Description |
|---|---|
| Operating mode | Operating mode state |
| Auto.-Addr.progr. executed | Automatic address assignment executed |
| Auto-Addr.progr. possible | Automatic address assignment possible |
| AS-i slave with addr.=0 available | Slave with address 0 available |
| Reference config. = actual config. | Target configuration equal to actual configuration |

| State | Marking |
|---|---|
| Slave OK | Green |
| Slave fault | Red |
| Slave is available, but not configured | Yellow |

## Procedure




1. Select the "Diagnostics" operating area.



2. Press the menu forward key twice and the "AS-i diag." softkey.
   The "Diagnostics AS-Interface" window is opened.




3. Select the required DP AS-interface.
   The complete diagnostics information is displayed.
   The values in italics indicate an error counter > 0.

# OEM-specific online help 16

## 16.1 Overview

In addition to the existing extensive online help, you also have the option of generating a manufacturer-specific online help and then linking this into SINUMERIK Operate.

This online help is generated in the HTML format, i.e. it comprises HTML documents that are linked with one another. The subject being searched for is called in a separate window from a contents or index directory. Similar to a document browser (e.g. Windows Explorer), a list of possible selections is displayed in the left-hand half of the window and when you click on the required subject, the explanation is displayed in the right hand half of the window.

Context sensitive selection of online help pages is not possible.

### Procedure

1. Generating HTML files
2. Generating a help book
3. Integrating the online help in SINUMERIK Operate
4. Saving help files

### Other application cases

Online help for the following OEM-specific expansions can be created and used to supplement the SINUMERIK Operate online help system:

- Online help for **cycles and/or M functions** of the machine manufacturer which extend the programming options for SINUMERIK control systems. This online help is called in just the same way as the SINUMERIK Operate online help "Programming".
- Online help for OEM-specific **variables** of the machine manufacturer. This online help is called from the variable view of SINUMERIK Operate.

### Programming online help

You can use the "SINUMERIK HMI programming package sl" for additional options for configuring the online help. Using this programming package, it is possible to develop high-level language applications in the C++ programming language for SINUMERIK Operate on the NCU 7x0.

**Note**

The "SINUMERIK HMI programming package sl" must be ordered separately as a software option. The associated documentation is provided together with the programming package.

## 16.2 Generating HTML files

Generating help files in the HTML format. It is possible to save all information in a single HTML file or to distribute the information over several HTML files.

You can assign the file names yourself, however, you must observe the following:

- References within HTML files should always be specified with relative paths. Only then can it be ensured that the references function in precisely the same way on both the development computer as well as on the target system.
- If jumps are to be made to certain points within an HTML file per link, then so-called anchors must be defined for this purpose.
  Example of an HTML anchor:
  `<a name="myAnchor">This is an anchor</a>`
- The contents of HTML documents must be saved with the UTF-8 coding. Only then is it guaranteed that the HTML documents are correctly displayed in all of the country languages supported by SINUMERIK Operate.
- The following sub-sets of the HTML functional scope are supported:

### HTML tags

| Tag | Description | Comment |
|---|---|---|
| a | Anchor or link | Supported attributes: href and name |
| address | Address | |
| big | Larger font | |
| block quote | Indented paragraph | |
| body | Document body | Supported attributes: bgcolor (#RRGGBB) |
| br | Line break | |
| center | Centered paragraph | |
| cite | Inline citation | Same effect as tag i |
| code | Code | Same effect as tag tt |
| dd | Definition data | |
| dfn | Definition | Same effect as tag i |
| div | Document division | The standard block attributes are supported |
| dl | Definition list | The standard block attributes are supported |
| dt | Definition term | The standard block attributes are supported |
| em | Emphasized | Same effect as tag i |
| font | Font size, family, color | Supported attributes: size, face, and color (#RRGGBB) |
| h1 | Level 1 heading | The standard block attributes are supported |
| h2 | Level 2 heading | The standard block attributes are supported |
| h3 | Level 3 heading | The standard block attributes are supported |
| h4 | Level 4 heading | The standard block attributes are supported |
| h5 | Level 5 heading | The standard block attributes are supported |
| h6 | Level 6 heading | The standard block attributes are supported |
| head | Document header | |

| Tag | Description | Comment |
|---|---|---|
| hr | Horizontal line | Supported attributes: width (can be specified as absolute or relative value) |
| html | HTML document | |
| i | Italic | |
| img | Image | Supported attributes: src, width, height |
| kbd | User-entered text | |
| meta | Meta information | |
| li | List item | |
| nobr | Non-breakable text | |
| ol | Ordered list | The standard attributes for lists are supported |
| p | Paragraph | The standard block attributes are supported (default setting: left-aligned) |
| pre | Preformatted text | |
| s | Strikethrough | |
| samp | Sample code | Same effect as tag tt |
| small | Small font | |
| span | Grouped elements | |
| strong | Strong | Continuous text is heavily emphasized, replaces tag b |
| sub | Subscript | |
| sup | Superscript | |
| table | Table | Supported attributes: border, bgcolor (#RRGGBB), cellspacing, cellpadding, width (absolute or relative), height |
| tbody | Table body | No effect |
| td | Table data cell | The standard attributes for table cells are supported |
| tfoot | Table footer | No effect |
| th | Table header cell | The standard attributes for table cells are supported |
| thead | Table header | This is used to print tables that extend over several pages |
| title | Document title | |
| tr | Table row | Supported attributes: bgcolor (#RRGGBB) |
| tt | Typewrite font | |
| u | Underlined | |
| ul | Unordered list | The standard attributes for lists are supported |
| var | Variable | Same effect as tag tt |

## Block attributes

The following attributes are supported by the tags div, dl, dt, h1, h2, h3, h4, h5, h6, p:

- align (left, right, center, justify)
- dir (ltr, rtl)

### Standard attributes for lists

The following attributes are supported by tags ol and ul:

- type (1, a, A, square, disc, circle)

### Standard attributes for tables

The following attributes are supported by tags td and th:

- width (absolute, relative, no-value)
- bgcolor (#RRGGBB)
- colspan
- rowspan
- align (left, right, center, justify)
- valign (top, middle, bottom)

### CSS properties

The following table includes the supported CSS functional scope:

| Property | Values | Description |
|---|---|---|
| background color | <color> | Background color for elements |
| background-image | <uri> | Background image for elements |
| color | <color> | Foreground color for text |
| text-indent | <length>px | Indent the first line of a paragraph in pixels |
| white-space | normal \| pre \| nowrap \| pre-wrap | Defines how whitespace characters are handled in HTML documents |
| margin-top | <length>px | Width of the upper edge of the paragraph in pixels |
| margin-bottom | <length>px | Width of the lower edge of the paragraph in pixels |
| margin-left | <length>px | Length of the left hand edge of the paragraph in pixels |
| margin-right | <length>px | Width of the right-hand edge of the paragraph in pixels |
| vertical-align | baseline \| sub \| super \| middle \| top \| bottom | Vertical alignment for text (in tables, only the values middle, top and bottom are supported) |
| border-color | <color> | Border color for text tables |
| border-style | none \| dotted \| dashed \| dot-dash \| dot-dot-dash \| solid \| double \| groove \| ridge \| inset \| outset | Border style for text tables |
| background | [ <'background-color'> \|\| <'background-image'> ] | Short notation for background property |
| page-break-before | [ auto \| always ] | Page break before a paragraph/table |
| page-break-after | [ auto \| always ] | Page break after a paragraph/table |
| background-image | <uri> | Background image for elements |

### Supported CSS selectors

All of the CSS 2.1 selector classes are supported, with the exception of so-called pseudo-selector classes, such as :first-child, :visited and :hover.

## 16.3 Generating the help book

The help book is an XML file in which the structure of the online help is defined. In this file, you define:

- HTML documents
- Contents and subject index

### Syntax for the help book

| Tag | Number | Meaning | |
|---|---|---|---|
| HMI_SL_HELP | 1 | Root element of the XML document | |
| I-BOOK<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I | + | Identifies a help book. The name can be freely selected under the constraint that no name predefined by the system is used (such as sinumerik_alarm_plc_pmc).<br>In the example, the name of the help book is "hmi_myhelp"<br>Attributes: | |
| | | ref | Identifies the HTML document that is displayed as the entry page for the help book. |
| | | title | Title of the help book that is displayed in the table of contents. |
| | | helpdir | Directory that contains the online help of the help book. |
| I-ENTRY<br>II<br>II<br>II<br>II<br>II<br>II<br>II | * | Section of the online help<br>Attributes: | |
| | | ref | Identifies the HTML document that is displayed as entry page for the section. |
| | | title | Title of the section that is displayed in the table of contents. |
| II-INDEX_ENTRY<br>II<br>II<br>II<br>II<br>II<br>II<br>II | * | Subject (keyword) to be displayed<br>Attributes: | |
| | | ref | Identifies the HTML document that is jumped to for this subject index entry. |
| | | title | Title of the subject that is displayed in the subject index. |

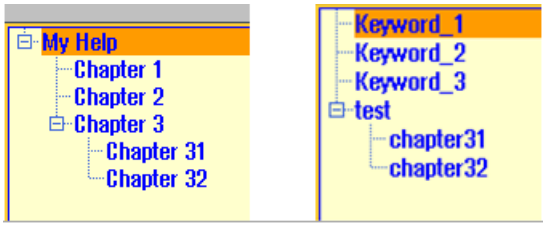
The following applies for the "Number" column:
* means 0 or more
+ means 1 or more

## Example for a help book

In the following example, the structure of a help book with the "My Help" name is described. Further, it forms the basis for the table of contents and subject index.

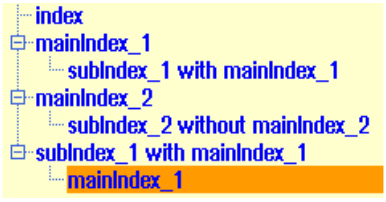
```
<?xml version="1.0" encoding="utf-8"?>
<HMI_SL_HELP language="en-US">
  <BOOK ref="index.html" title="My Help" helpdir="hmi_myhelp">
    <ENTRY ref="section_1.html" title="Section 1">
      <INDEX_ENTRY ref="section_1html#Keyword_1" title="Keyword_1"/>
      <INDEX_ENTRY ref="section_1.html#Keyword_2" title="Keyword_2"/>
    </ENTRY>
    <ENTRY ref="section_2.html" title="Section 2">
      <INDEX_ENTRY ref="section_2.html#Keyword_3" title="Keyword_3"/>
    </ENTRY>
    <ENTRY ref="section_3.html" title="Section 3">
      <ENTRY ref="section_31.html" title="Section 31">
        INDEX_ENTRY ref="section_31.html#test" title="test;section31"/>
      </ENTRY>
      <ENTRY ref="section_32.html" title="Section 32">
        INDEX_ENTRY ref="section_32.html#test" title="test;section32"/>
      </ENTRY>
    </ENTRY>
  </BOOK>
</HMI_SL_HELP>
```

The book comprises three sections, whereby the third section has two subsections. The various subject words (keywords) are defined within the section.



You have the following three options to format the subject index:

1. Single entry:
   `<INDEX_ENTRY ...title="index"/>`
2. Two two-stage entry, whereby each title has a main and a subentry. Separate the entries from one another using a comma.
   `<INDEX_ENTRY ...title="mainIndex_1,subIndex_1 with mainIndex_1"/>`
3. Two-stage entry, whereby the first title is the main entry and the second title is the subentry. Separate the entries from one another using a semicolon.
   `<INDEX_ENTRY ...title="mainIndex_2;subIndex_2 without mainIndex_1"/>`




## 16.4 Integrating the online help in SINUMERIK Operate

If you wish to integrate the generated help book into the online help system of SINUMERIK Operate, then you require the "slhlp.xml" file.

### Format description of the "slhlp.xml"

| Tag | Num-ber | Meaning | |
|---|---|---|---|
| CONFIGURATION | 1 | Root element of the XML document. Designates that this involves a configuration file. | |
| I-OnlineHelpFiles | 1 | Introduces the section about the online help books. | |
| II-<help_book> | * | Introduces the section of a help book. | |
| III-EntriesFile<br>III<br>III<br>III<br>III<br>III | 1 | File name of the help book with the list of contents and subject (keyword) entries.<br>Attributes: | |
| | | value | Name of the XML file |
| | | type | Data type of the value (QString) |
| III-Technology<br>III<br>III<br>III<br>III<br>III<br>III<br>III<br>III<br>III<br>IIII | 0, 1 | Specifies the technology that applies to the help book.<br>"All" applies to all technologies.<br>If the help book applies to several technologies, then the technologies are listed separated by comma.<br>Possible values:<br>All, Universal, Milling, Turning, Grinding, Stroking, Punching<br>Attributes: | |
| | | value | Technology data |
| | | type | Data type of the value (QString) |

| Tag | Num-ber | Meaning | |
|---|---|---|---|
| III -DisableSearch<br>III<br>III<br>III<br>III<br>III | 0, 1 | Disable the subject (keyword) search for the help book.<br>Attributes: | |
| | | value | true, false |
| | | type | type, data type of the value (bool) |
| III-DisableFullTextSearch<br>III<br>III<br>III<br>III | 0, 1 | Disable the full text search for the help book.<br>Attributes: | |
| | | value | true, false |
| | | type | type, data type of the value (bool) |
| III-DisableIndex<br>III<br>III<br>III<br>III | 0, 1 | Disable the subject index for the help book.<br>Attributes: | |
| | | value | true, false |
| | | type | type, data type of the value (bool) |
| III-DisableContent<br>III<br>III<br>III<br>III | 0, 1 | Disable the table of contents for the help book.<br>Attributes: | |
| | | value | true, false |
| | | type | type, data type of the value (bool) |
| III-DefaultLanguage<br>III<br>III<br>III<br>III<br>III | 0, 1 | Abbreviation for the language that should be displayed if the actual country language is available for the help book.<br>Attributes: | |
| | | value | chs, deu, eng, esp, fra, ita, ... |
| | | type | Data type of the value (QString) |

The following applies for the "Number" column:
* means 0 or more

## Example of a file "slhlp.xml"

The help book "hmi_myhelp.xml" is made known to SINUMERIK Operate in the following example.

The subject index has not been activated for the help book.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE CONFIGURATION>
<CONFIGURATION>
   <OnlineHelpFiles>
     <hmi_myHelp>
       <EntriesFile value="hmi_myhelp.xml" type="QString"/>
       <DisableIndex value="frue" type="bool"/>
     </hmi_myHelp>
   </OnlineHelpFiles>
</CONFIGURATION>
```

## 16.5 Saving help files

### Saving help files in the target system

1. Open the /**oem**/sinumerik/hmi/hlp directory and create a new folder for the required language. For this purpose, use the specified language code.
   It is mandatory that the folder names are written in lower-case letters.
   For instance, if you are integrating a help function for German and English, then create the "deu" and "eng" folders.
2. Place the help book, e.g. "hmi_myhelp.xml" in the "deu" and "eng" folders.
3. Copy the help files into the directories, e.g. /**oem**/sinumerik/hmi/hlp/deu/hmi_myhelp for German and /**oem**/sinumerik/hmi/hlp/eng/hmi_myhelp for English help files.
4. Place the configuration file "slhlp.xml" into the directory /**oem**/sinumerik/hmi/cfg.
5. Restart the HMI.

**Note**

When displaying the list of contents and subject index of a help book, the help files are saved in the binary format (slhlp_<Hilfe-Buch_*.hmi) under the directory /siemens/sinumerik/sys_cache/hmi/hlp for faster use. If you change the help book, you must always delete these files.

### See also

Supported languages (Page 502)

## 16.6 Generating online help for user alarms and machine data

You have the option to create a dedicated online help for user-specific PLC alarms. These can be opened context-sensitive from the alarm list when the alarms occur.

The help texts for the user-specific PLC alarms are realized in the HTML file "sinumerik_alarm_oem_plc_pmc.html".

### Creating an HTML file

1. Create the HTML file in the following directory: /**oem**/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_plc_pmc/ or /**user**/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_plc_pmc/.
   <lng> stands for the language code, e.g. deu, eng, etc. If you create a language directory, use the language code from the table in Section Supported languages (Page 502)
2. Assign the "sinumerik_alarm_oem_plc_pmc.html" name to the file. If you create help texts for several languages, then create the corresponding number of HTML files with precisely these names. This file name must not be changed and is applicable for all languages.

3. Place the files in the corresponding language directories. For instance, place the file for the German help text in the following directory: /**oem**/sinumerik/hmi/hlp/deu/ sinumerik_alarm_plc_pmc/sinumerik_alarm_oem_plc_pmc.html
4. Delete the file "slhlp_sinumerik_alarm_plc_pmc_*.hmi" in the /**siemens**/sinumerik/ sys_cache/hmi//hlp directory and restart the operating software.

### "sinumerik_alarm_oem_plc_pmc.html" file

| Entry | Meaning |
|---|---|
| <a name="AlarmNr">AlarmNr</a> | Hyperlink to the alarm number |
|  ..... | Help text for the corresponding alarm |
| <td width="85%">.....</td> | Text that is displayed after the "Explanation" or "Remedy" field. |

## Example

The alarm number is used as HTML anchor.

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE html PUBLIC "-//W3C//DTD
HTML 4.0 Transitional//EN" >
- <html>
   <head><title></title></head>
-  <body>
-    <table>
-      <tr>
-        <td width="15%">
           <b><a name="510000">510000</a></b>
         </td>
-        <td width="85%">
           <b>This is the help for user alarm 510000</b>
         </td>
       </tr>
-      <tr>
-        <td valign="top" width="15%">
           <b>Explanation</b>
         </td>
         <td width="85%"> This is the explanation of user alarm 510000.</td>
       </tr>
-      <tr>
         <td valign="top" width="15%"><b>Remedy:</b></td>
         <td width="85%">Eliminate reason for alarm. </td>
       </tr>
     </table>

   </body>
 </html>
```

## Distribute the online help for user alarms over several HTML files

You have the option of distributing the online help for user alarms over several HTML files.

1. Create an XML file with the name "sinumerik_alarm_plc_pmc.xml" in the following directory: /**oem**/sinumerik/hmi/hlp/<lng>/ or /**user**/sinumerik/hmi/hlp/<lng>/
   <lng> stands for the language code, e.g. deu, eng, etc. When you create a language directory, use the language code from the table in Section Supported languages (Page 502)
2. Open the XML file and in the tag <NUM_AREAS> enter the number ranges of the alarms and the corresponding HTML files.
3. Store the HTML files, specified in the XML file, in the following directory: /**oem**/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_plc_pmc or /**user**/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_plc_pmc file

## Example: Creating an XML file

| Entry | Meaning |
|---|---|
| <BOOK> | Help book |
| <NUM_AREAS> | Number ranges of the alarms and the reference to the particular HTML file |

The following example includes the online helps for the
alarms from 500000 to 599999 in the HTML file "sinumerik_alarm_oem_1.html",
the alarms from 600000 to 699999 in the HTML file "sinumerik_alarm_oem_2.html",
the alarms from 700000 to 799999 in the HTML file "sinumerik_alarm_oem_3.html",
the alarms from 800000 to 899999 in the HTML file "sinumerik_alarm_oem_4.html".

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
  <!DOCTYPE HMI_SL_HELP>
  <HMI_SL_HELP>
   <BOOK>
     <NUM_AREAS>
       <NUM_AREA from="500000" to="599999" ref="sinumerik_alarm_oem_1.html" />
       <NUM_AREA from="600000" to="699999" ref="sinumerik_alarm_oem_2.html" />
       <NUM_AREA from="700000" to="799999" ref="sinumerik_alarm_oem_3.html" />
       <NUM_AREA from="800000" to="899999" ref="sinumerik_alarm_oem_4.html" />
     </NUM_AREAS>
   </BOOK>
  </HMI_SL_HELP>
```

## Replacing the standard online help

You have the option of generating your own online help for the HMI, NC and PLC alarms as well as machine data. This means that you can replace the standard online help by your own online help.

You can call your own online help directly at the control using the INFO key. Instead of the standard online help, your own online help is displayed.

**Directory for HTML files**

| Area | Directory |
|---|---|
| PLC alarms | /**oem**/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_plc_pmc/ or /**user**/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_plc_pmc/ |
| HMI alarms | /**oem**/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_hmi/ or /**user**/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_hmi/ |
| NC alarms | /**oem**/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_nck/ or /**user**/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_nck/ |
| NC MD | /**oem**/sinumerik/hmi/hlp/<lng>/sinumerik_md_nck/ |
| Channel MD | /**oem**/sinumerik/hmi/hlp/<lng>/sinumerik_md_chan/ |
| Axis MD | /**oem**/sinumerik/hmi/hlp/<lng>/sinumerik_md_axis/ |

<lng> stands for the language code, e.g. deu, eng, etc. Only use the language codes from the table in SectionSupported languages (Page 502)

**Procedure:**

1. Create the corresponding directory, refer to the table "Directory for HTML files".
2. Generate the help file and as file names, use the alarm/machine data number as well as the extension ".html" for HTML help files and ".pdf" for PDF help files. If you create help texts for several languages, then create the corresponding number of help files with precisely these names and then create the file in the corresponding language directory.

## Examples:

- Your own German online help for the PLC Alarm 510000:
  /oem/sinumerik/hmi/hlp/deu/sinumerik_alarm_plc_pmc/510000.html
  /oem/sinumerik/hmi/hlp/deu/sinumerik_alarm_plc_pmc/510000.pdf
- Your own German online help for the NC machine data 14510 $MN_USER_DATA_INT:
  /oem/sinumerik/hmi/hlp/deu/sinumerik_md_nck/14510.html

# 16.7 Example: This is how you create an online help for NC/PLC variables

## Overview

In order to create context-sensitive online help for NC/PLC variables or system variables as shown in the following example, the descriptive texts are managed in language-dependent html files.

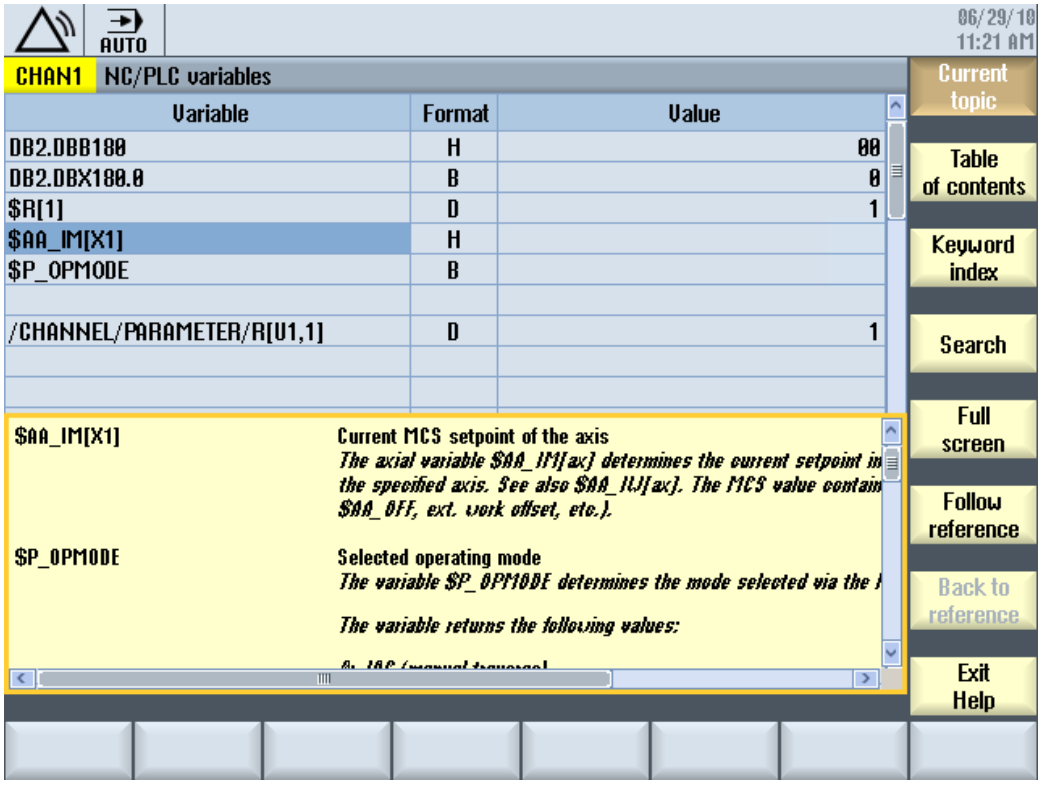



Figure 16-1 Example: Online help for user variables

## Structure of the online help

The following files are required for the online help:

| File | Meaning |
|---|---|
| sldgvarviewhelp.ini | Configuration file to manage an html file or several html files |
| <lng>/<name>1.html | The contents of all html files of the online help are language-dependent and are saved in the relevant language directory <lng>. |
| <lng>/<name>2.html | |
| . . . | |
| <lng>/<name>n.html | |

## Structure of the configuration file

The file is **independent of any language** and is structured as follows:

**sldgvarviewhelp.ini**

```
[HelpBindings]
/BAG/STATE/OPMODE = var1_help.html#var1
$AA_IM[X1] = var1_help.html
$R[1] = var1_help.html#var2
/Channel/Parameter/R[u1,1] = var2_help.html#var2
DB2.DBX180.0 = var2_help.html#var1
GUD/MyVar[2] = var2_help.html
```

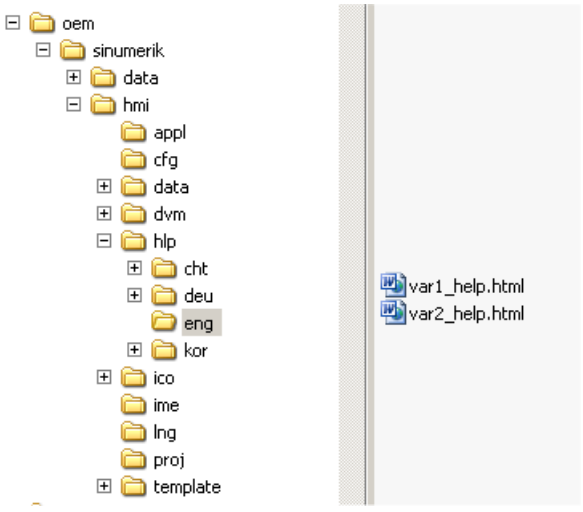
### Note

The html files can be generated with any html editor. A definition is provided in the configuration file as to which html files belong to the online help.

The description can comprise one or several html files: For example, one html file for each variable or several identical variables in one file.

### Procedure:

1. Copy the configuration file to the following directory:
   `/oem/sinumerik/hmi/cfg/sldgvarviewhelp.ini`
2. Copy the html files to one of the following directories:
   `/oem/sinumerik/hmi/hlp/<lng>/`
   `/user/sinumerik/hmi/hlp/<lng>/`
   <lng> stands for the language code.




3. Delete all files in the directory:
   `/siemens/sinumerik/sys_cache/hmi//hlp`

The settings become effective only after restarting the system.

# 16.8 Example: This is how you create a programming online help

## Preconditions

Create the following files:

- Configuration file "prog_help.ini"

```
[milling]
CYCLE1=cycle1_help.html
CYCLE2=cycle2_help.html#TextAnchor1
CYCLE3=cycle3_help.html
CYCLE4=cycle4_help.html

[turning]
CYCLE3=cycle2_help.html
CYCLE4=cycle3_help.html
```

- Configuration file for the help book "slhlp.xml" (optional)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE CONFIGURATION>
<CONFIGURATION>
  <OnlineHelpFiles>
    <hmi_prog_help>
      <EntriesFile value="hmi_prog_help.xml" type="QString"/>
      <DisableIndex value="true" type="bool"/>
    </hmi_prog_help>
  </OnlineHelpFiles>
</CONFIGURATION>
```

- Configuration file for the help book "hmi_prog_help.xml" (optional)

```
<?xml version="1.0" encoding="UTF-8"?>
<HMI_SL_HELP language="en-US">
  <BOOK ref="index.html" title="OEM_CYCLES" helpdir="hmi_prog_help">
    <ENTRY ref="cycle1_help.html" title="Cycle1"></ENTRY>
    <ENTRY ref="cycle2_help.html" title="Cycle2"></ENTRY>
    <ENTRY ref="cycle3_help.html" title="Cycle3"></ENTRY>
    <ENTRY ref="cycle4_help.html" title="Cycle4"></ENTRY>
    <ENTRY ref="cycle_help.html" title="OEM_Cycles"></ENTRY>
  </BOOK>
</HMI_SL_HELP>
```

- Language-dependent file "<prog_help_eng>.ts": this filename is permanently specified.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!DOCTYPE TS>
<TS>
    <context>
      <name>oem_cycles</name>
        <message>
          <source>CYCLE1</source>
          <translation>short description for cycle 1</translation>
          <chars>*</chars>
        </message>
        <message>
          <source>CYCLE2</source>
          <translation>short description for cycle 2</translation>
          <chars>*</chars>
        </message>
        <message>
          <source>CYCLE3</source>
          <translation>short description for cycle 3</translation>
          <chars>*</chars>
        </message>
    </context>
</TS>
```

### Saving help files in the target system

Procedure:

1. Copy the configuration file "prog.ini" to the following directory:
   `/oem/sinumerik/hmi/cfg`
2. Copy the file "`slhlp.xml`" to the help book in the following directory:
   `/oem/sinumerik/hmi/cfg`
3. Create a directory for the desired language of the online help under the following path: `/oem/sinumerik/hmi/hlp/<lng>` and copy the file "`hmi_prog_help.xml`" to this directory
   Use the specified language code from the list of language codes for file names. The directory names must be written in lower case.
4. Copy the language-dependent `prog_help_<lng>.ts` file for the Product Brief to the following path: `/oem/sinumerik/hmi/<lng>/prog_help_<lng>.ts`
5. Copy the html files with the description of the OEM cycles to the following directory:
   `/oem/sinumerik/hmi/hlp/<lng>/hmi_prog_help/cycle<n>_help.html`

The settings become effective only after restarting the system.

# 16.9 Help files in PDF format

In addition to help files in HTML format, you can also include PDF-format information in the operating software. Individual PDF helps can be opened with links from the table of contents or index, or directly from HTML files.

## Storing PDF helps

Copy the PDF helps to one of the following directories:

```
/oem/sinumerik/hmi/hlp/<lng>/<hmi_myhelp>
```

```
/user/sinumerik/hmi/hlp/<lng>/<hmi_myhelp>
```

## Including PDF helps

Bind in dialog configurations or configurations of table of contents and indexes with the extension "pdf" in the same manner as for the "html" extension:

```
<ENTRY ref="myFile.pdf" title="Help 1">
```

Create a link from HTML files to the PDF help:

```
<a href="myFile.pdf">My Help File</a>
```

**Note**

It is not possible to select context-sensitive jump labels in the PDF help, select jump labels or make jumps to other HTML or PDF files.

The search function is possible only within a PDF file. A higher-level search over multiple PDF helps is not supported.

# SINUMERIK Integrate - AMB, AMC, AMM 17

## Precondition

To start the "SINUMERIK Integrate - AMB, AMC, AMM" software, configure a softkey using a display machine data. To do this you require, as a minimum, authorization for access level 3 (user).

## Configuring a softkey

| MD9108 $MM_SINUMERIK_INTEGRATE | |
|---|---|
| = 1 | The "SINUMERIK integrate" softkey is displayed on the extended user interface bar. |

## References

Information on the software can be found in the following documentation:

Function Manual SINUMERIK Integrate - AMB, AMC, AMM

# Technologies and cycles 18

## 18.1 Activating turning/milling/drilling/grinding technologies

The cycles are automatically loaded when the controller boots (Power On).

The configuration is realized using cycle machine data and cycle setting data. In so doing, a differentiation is made between the following data:

- General cycle machine data
- Channel-specific cycle machine data
- Axis-specific cycle machine data
- General cycle setting data
- Channel-specific cycle setting data
- Axis-specific cycle setting data

### Requirement for memory configuration

To ensure that the cycles run correctly, the following standard setting must be checked:

| MD28082 $MC_MM_SYSTEM_FRAME_MASK | System frames (SRAM) |
|---|---|
| Bit 5 | System frame for cycles |
| = 1 | Default setting |

| MD52201 $MCS_TECHNOLOGY_EXTENSION | Technology extension for combined machines with several technologies |
|---|---|
| = 1 | Additional settings for turning |
| = 2 | Additional settings for turning,<br>e.g. turnign machine with milling technology<br>MD52200 $MCS_TECHNOLOGY = 1<br>MD52201 $MCS_TECHNOLOGY_EXTENSION = 2 |

**Note**

Cylindrical and surface grinding are not supported as a technology extension by SINUMERIK Operate.

## Adapting the user interface

The softkey display for the selection of the cycle support in the program editor is set using the following channel-specific configuration machine data:

| MD52200 $MCS_TECHNOLOGY | Technology |
|---|---|
| = 1 | Setting for turning |
| = 2 | Setting for milling |
| = 3 | Setting for the cylindrical grinding technology |
| = 4 | Setting for the surface grinding technology |

### References: Grinding

For further information about the grinding technology, please refer to:

- Chapter "Grinding (Page 401)"
- Operating Manual, Grinding
- Programming Manual, Production Planning; Chapter "Programming cycles externally" > "Technological cycles":
  - CYCLE495 - form-truing
  - CYCLE4071 - CYCLE4074: Longitudinal grinding cycles
  - CYCLE4075 - CYCLE4079: Surface grinding cycles
- Function Manual, Extended Functions; Compensations (K3),
  Section: "Interpolatory compensation" > "Cylinder error compensation"

## Configuring the screen display

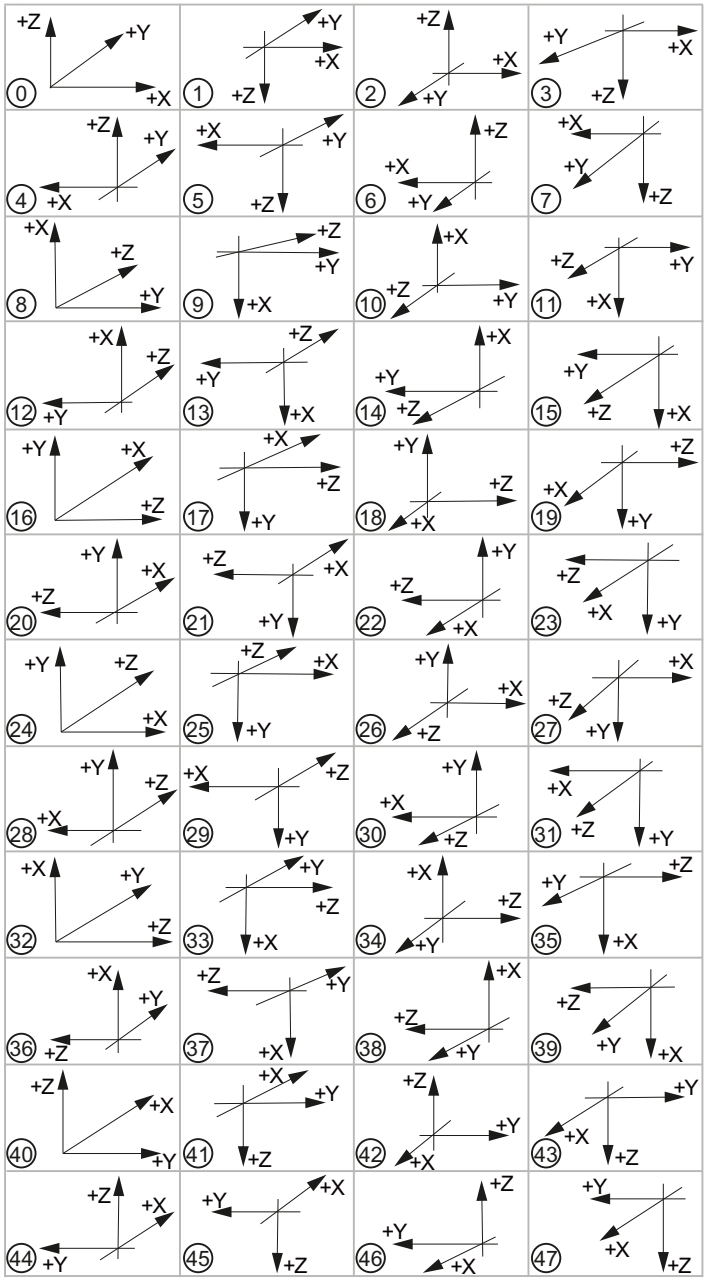
The screen display can be configured using the following channel-specific machine data:

| MD52210 $MCS_FUNCTION_MASK_DISP | Function screen display |
|---|---|
| Bit 0 | Dimension system for programs always in the basic system |
| Bit 1 | Front view when turning in the school coordinate system |
| Bit 2 | Hide "T,S,M" softkey in the JOG area |
| Bit 3 | Generate end of program automatically in MDI (with "Delete blocks" softkey) |
| Bit 4 | Display follow-on tool in the T,F,S window |
| Bit 5 | Hide "MCS " softkey |
| Bit 6 | Hide tool radius/diameter in the T,F,S window |
| Bit 7 | Hide tool lengths in the T,F,S window |
| Bit 8 | Hide tool icon in the T,F,S window |
| Bit 9 | Display "Easy" input selection, see Section ""Easy" input in cycle support" |
| Bit 10 | Do not offer channel in job lists |
| Bit 11 | For WCS, activate the display of the logical spindles |
| Bit 12 | Hide mold making view for G code |

## Setting the coordinate system

| MD52000 $MCS_DISP_COORDINATE_SYSTEM | Position of the coordinate system |
|---|---|
| = 0 | Default setting |
| With this machine data you adapt the position of the coordinate system of the user interface to the coordinate system of the machine. In the user interface, all help screens, sequence graphics, simulations, and input fields with circle direction indication change automatically according to the selected position. | |

The following settings can be selected for the coordinate system:



## Meaning of the axes

| MD52206 $MCS_AXIS_USAGE[n] | Meaning of the axes in the channel<br>[n] channel axis number |
|---|---|
| = 0 | No special meaning |
| = 1 | Tool spindle (rotating tool) |
| = 2 | Auxiliary spindle (rotating tool) |
| = 3 | Main spindle (turning) |
| = 4 | Separate C axis of the main spindle (turning) |
| = 5 | Counterspindle (turning) |
| = 6 | Separate C axis of the counterspindle (turning) |
| = 7 | Linear axis of the counterspindle (turning) |
| = 8 | Tailstock (turning) |
| = 9 | Steady (turning) |
| = 10 | B axis (turning) |
| = 12 | B axis in counterspindle (turning) |
| = 13 | Transverse travel X of the counterspindle (turning) |

## Defining the direction of rotation

Enter the direction of rotation for the rotary axes that are not configured in a tool holder or a 5-axis transformation via the following channel-specific machine data.

| MD52207 $MCS_USAGE_ATTRIB[n] | Attribute of the axes in the channel<br>[n] channel axis number |
|---|---|
| Bit 0 | Rotates around 1st geometry axis (for rotation axis) |
| Bit 1 | Rotates around 2nd geometry axis (for rotation axis) |
| Bit 2 | Rotates around 3rd geometry axis (for rotation axis) |
| Bit 3 | Direction of rotation is counter-clockwise (for rotary axis / C axis). |
| Bit 4 | Displayed direction of rotation of the spindle/C axis for the M function M3 is counter-clockwise |
| Bit 5 | Inverts M3 / M4 (for spindles)<br>This bit must be set analogous to PLC bit DB31, ... DBX17.6! |
| Bit 6 | Display rotary axis as offset target for measurement |
| Bit 7 | List rotary axis in the position pattern |
| Bit 8 | List rotary axis to accept blank (on milling machines) |
| Bit 9 | Spindle is not SPOS-capable |
| Bit 10 | Rotary axis rotates around 1st geometry axis (only for position pattern) |
| Bit 11 | Rotary axis rotates around 2nd geometry axis (only for position pattern) |
| Bit 12 | Rotary axis rotates around 3rd geometry axis (only for position pattern) |

## "Easy" input in cycle support

You can influence the number of visible input parameters via MD52210 $MCS_FUNCTION_MASK_DISP with bit 9:

- All input parameters are displayed with the default setting bit 9 = 0 for "Complete".
- With the setting bit 9 = 1, you can select "Complete" or "Easy" input parameters in the cycle support.

If you select the "Easy" input screen for input of the parameters, the values from the following setting data are used for the invisible parameters (can be adapted machine-specifically):

| SD55300 $SCS_EASY_SAFETY_CLEARANCE | Easy input: Safety clearance |
|---|---|
| = 1 | mm |

| SD55301 $SCS_EASY_DWELL_TIME | Easy input: Dwell time |
|---|---|
| = 0.6 | s |

| SD55305 $SCS_EASY_DRILL_DEEP_FD1 | Easy input: Percentage of 1st deep-hole drilling feedrate |
|---|---|
| = 90 | % |

| SD55306 $SCS_EASY_DRILL_DEEP_DF | Easy input: Percentage of deep-hole drilling infeed |
|---|---|
| = 90 | % |

| SD55307 $SCS_EASY_DRILL_DEEP_V1 | Easy input: Min. deep-hole drilling depth infeed |
|---|---|
| = 1.2 | mm |

| SD55308 $SCS_EASY_DRILL_DEEP_V2 | Easy input: Deep-hole drilling retraction distance |
|---|---|
| = 1.4 | mm |

| SD55309 $SCS_EASY_THREAD_RETURN_DIST | Easy input: Thread turning return distance |
|---|---|
| = 2 | mm |

**Note**

The specified default values are valid for the metric system and are converted by the NC automatically on the machine with the inch setting.

### Additional settings

| MD52005 $MCS_DISP_PLANE_MILL | Plane selection G17, G18, G19 |
|---|---|
| = 0 | Milling: Plane selection in the cycle support when programming under "programGUIDE G code" |
| = 17 | G17 plane (default value) |
| = 18 | G18 plane |
| = 19 | G19 plane |

| MD52006 $MCS_DISP_PLANE_TURN | Plane selection G17, G18, G19 |
|---|---|
| = 18 | G18 plane (permanently set) |

| MD52212 $MCS_FUNCTION_MASK_TECH | Cross-technology function screen |
|---|---|
| Bit 0 | Enable swivel |
| = 0 | Swivel plane, swivel tool not enabled |
| = 1 | Swivel plane, swivel tool enabled |
| Bit 1 | No optimized travel along software limit switches |
| = 0 | No optimized travel along software limit switches |
| = 1 | Optimized travel along software limit switches |

| MD52240 $MCS_NAME_TOOL_CHANGE_PROG | Tool change program for G code steps |
|---|---|
| = Program name | The associated program is called for tool change. |

### See also

Technology cycles for swiveling (Page 404)

## 18.2 Technology cycles for drilling

### Drilling technology

You can set drilling technology using the following channel-specific configuration machine data and channel-specific cycle setting data.

| MD52216 $MCS_FUNCTION_MASK_DRILL | Drilling function screen form |
|---|---|
| Bit 0 | Tapping cycle CYCLE84, technology input fields |
| = 0 | Hide input fields |
| = 1 | Display input fields |
| Bit 1 | Tapping cycle CYCLE840, technology input fields |
| = 0 | Hide input fields |
| = 1 | Display input fields |

| SD55216 $SCS_FUNCTION_MASK_DRILL_SET | Drilling function screen form |
|---|---|
| Bit 1 | Boring CYCLE86: Take into account the rotation of the tool plane when positioning the spindle<br>Note:<br>Spindle direction of rotation M3/ M4 and direction of rotation of the rotary axes must be set according to DIN. For SPOS=0, the tool cutting edge points in the + direction of the 1st axis of the plane (for G17, to X+). |
| = 0 | Do not take into account the rotation of the tool plane when positioning the spindle (SPOS). |
| = 1 | Take into account the rotation of the tool plane when positioning the spindle (SPOS). |
| Bit 2 | Boring CYCLE86: When positioning the spindle, take into account swiveled table kinematics (tool carrier) |
| = 0 | When positioning the spindle, do not take into account the components of the swiveled table kinematics (swiveling with CYCLE800, tool carrier).<br>Note:<br>Only the rotary axis of the machine kinematics that rotates the workpiece is taken into account. In the initial state of the kinematics, this rotary axis must be orientated parallel to the tool axis.<br>e.g. rotary axis C rotation around Z (for G17) rotary axis vector of the swivel data set V2xyz = 0 , 0 , -1. |
| = 1 | When positioning the spindle, take into account the components of the swiveled table kinematics (swiveling with CYCLE800, tool carrier). |
| Bit 3 | Tapping CYCLE84: Monitoring of the machine data MD31050 $MA_DRIVE_AX_RATIO_DENOM and MD31060 $MA_DRIVE_AX_RATIO_NUMERA of the spindle |
| Bit 4 | Tapping CYCLE84, set monitoring of the machine data MD31050 $MA_DRIVE_AX_RATIO_DENOM and MD31060 $MA_DRIVE_AX_RATIO_NUMERA of the spindle |
| = 0 | No monitoring |
| = 1 | Monitoring |
| Bit 6 | Boring CYCLE86: Adapt the spindle position to the tool orientation<br>Tool orientation can be realized using TOOLCARR or TRAORI before the cycle is called. |
| = 1 | Bits 1 and 2 are no longer effective. |
| Bit 7 | Boring CYCLE86: Adapt the direction of retraction travel in the plane to the active mirroring |
| = 0 | Retraction in the plane is realized as programmed. |
| = 1 | When mirroring is active, the direction when retracting in the plane is adapted in the cycle to the active mirroring. |

## Tapping (CYCLE84 and CYCLE840)

If the technology screens are hidden using the channel-specific configuration machine data MD52216 $MCS_FUNCTION_MASK_DRILL, then the settings in the following channel-specific cycle setting data are effective:

| SD55481 $SCS_DRILL_TAPPING_SET_GG12[0] | Setting tapping G group 12:<br>Block change behavior for exact stop |
|---|---|
| = 0 | Exact stop response as before the cycle call (default value). |
| = 1 | G601 |
| = 2 | G602 |
| = 3 | G603 |

| SD55482 $SCS_DRILL_TAPPING_SET_GG21[0] | Setting tapping G group 21: Acceleration behavior |
|---|---|
| = 0 | Acceleration behavior as before the cycle call (default value). |
| = 1 | SOFT |
| = 2 | BRISK |
| = 3 | DRIVE |

| SD55483 $SCS_DRILL_TAPPING_SET_GG24[0] | Setting tapping G group 24: Precontrol |
|---|---|
| = 0 | Precontrol, the same as before the cycle call (default value). |
| = 1 | FFWON |
| = 2 | FFWOF |

### Tapping (CYCLE84)

| SD55484 $SCS_DRILL_TAPPING_SET_MC[0] | Spindle operation for MCALL |
|---|---|
| = 0 | For MCALL, reactivate spindle operation (default value). |
| = 1 | For MCALL, remain in position controlled spindle operation. |

# 18.3 Manufacturer cycles

## 18.3.1 Manufacturer cycles

The following cycles are also available in the cycle package for individual adaptation:

| | |
|---|---|
| CUST_TECHCYC.SPF | Manufacturer cycle for the adaptation of technology cycles. |
| CUST_800.SPF | Manufacturer cycle for the adaptation of the swivel plane and the swivel tool functions (CYCLE800), see Chapter "Manufacturer cycle CUST_800.SPF (Page 435)". |
| CUST_832.SPF | Manufacturer cycle for the adaptation of the high-speed settings function (CYCLE832)", see Chapter Manufacturer cycle CUST_832.SPF (Page 460)". |
| CUST_MEACYC.SPF | Manufacturer cycle for the adaptation of the measuring functions, see Chapter "Manufacturer and user cycle CUST_MEACYC.SPF (Page 466)". |
| CUST_MEAPROT.SPF | Manufacturer cycle and user cycle for the adaptation of the log form and content, see Chapter "Manufacturer and user cycle CUST_MEAP-ROT.SPF (Page 496)". |

| | |
|---|---|
| PROG_EVENT.SPF | Standard cycle to support the following functions:<br>• Block search when milling or turning is activated<br>• Block search and swivel plane<br>• Block search and align / advance tool<br>Note MD52212 $MCS_FUNCTION_MASK_TECH bits 3, 5 and 15 ... 17. |
| CUST_T | Cycle is used to track the T preparation after SERUPRO. |
| CUST_M6 | Cycle is used to track the tool change after SERUPRO. |
| CUST_MULTICHAN | Manufacturer cycle for lathes with several channels |
| CUST_CLAMP | Manufacturer's cycle for clamping elements |

### Copying manufacturer cycles

Copy the cycle from the NC data/Cycles/Standard cycles directory to the NC data/Cycles/User cycles directory before making any changes. Changes for the cycle take effect only in this directory.




1. Select the "Start-up" operating area.




2. Press the "System data" softkey.
3. Open the directory NC data/Cycles/Standard cycles and select the manufacturer cycles.




4. Press the "Copy" softkey and open the NC data/Cycles/Manufacturer cycles directory.




5. Press the "Paste" softkey.
   The cycles are available for you to individually adapt them.

**Note**

When upgrading the software, please refer to the upgrade instructions.

## 18.3.2 Prerequisites for the simulation mode

### Configuration

If you use feedback signals regarding the machine state, e.g. in CYCPE1MA, for tool change cycles, measuring cycles or CUST_800, then in the NC program you must link in the following system variables in order to avoid PLC access. Otherwise, PLC access prevents data alignment, results in alarms or the simulation being canceled:

| System variable | Meaning |
|---|---|
| $P_SIM | NCK simulation active |

### Example

In the following example, the simulation mode in PROG_EVENT or CYCPE1MA.SPF is stopped by the PLC access:

```
N100 REPEAT
N110 UNTIL PLC_VAR <> 0
```

If you use the system variable $P_SIM, then PLC access in the simulation mode is prevented:

```
N099 IF NOT $P_SIM
N100 REPEAT
N110 UNTIL PLC_VAR <> 0
N111 ENDIF
```

## 18.3.3 Extending the PROG_EVENT standard cycle

If you wish to extend the functionality of the PROG_EVENT, then you must not copy this into the manufacturer cycles directory, but you must create two new manufacturer cycles. Use the names CYCPE1MA.SPF or CYCPE_MA.SPF names for the manufacturer cycles.

For the manufacturer cycles CYCPE1MA.SPF or CYCPE_MA.SPF, the corresponding

jump markers are prepared at the beginning and at the end of PROG_EVENT.SPF.

If the CYCPE1MA.SPF and CYCPE_MA.SPF manufacturer cycles are created in the NC, the program correspondingly branches from PROG_EVENT to the manufacturer cycles.

### Procedure




1. Select the "Start-up" operating area.




2. Press the "System data" softkey.
3. Select the directory /NC data/Cycles/Manufacturer cycles




4. Press the "New" softkey.
   The "New G Code Program" window opens.
5. Assign the name CYCPE1MA or CYCPE_MA.
   Only use these names.




6 Press the "OK" softkey.
   The cycle has been created and you can now adapt it.

## 18.3.4 Manufacturer cycle for tool change CUST_T and CUST_M6

### Function

The two cycles are used to track the T preparation (CUST_T) and the tool change (CUST_M6) according to SERUPRO. In so doing, depending on the specific tool holder, the following is output:

- Last programmed change
- Last programmed preparation

### Precondition

The cycles can only be used when tool management is active.

### Sequence

The two cycles are called in the "SERUPRO-END-ASUB", which also provides the two transfer parameters.

| Parameter | Meaning |
|---|---|
| _THnr | Tool holder or master spindle number that was programmed for the change or the preparation. |
| _MTHnr | Number of the active tool holder or master spindle at the time of programming. |

The cycles themselves do not contain and "machine logic". Pure data operations are performed.

The active tool holder or the master spindle is saved.

Using the transfer parameter, the situation at the time of the T or M6 programming is determined. This situation is established, T and/or M06 output or a branch is made into the manufacturer cycles. After the output of T and M06 or after returning from the manufacturer cycles, the tool holder, saved at the beginning, is reactivated.

A change cycle can then only be called without any restriction, if the corresponding queries such as program test, block search,...exist.

### T replacement cycle

If a T replacement cycle is being used, then the following note must be observed:

**Note**

The CUST_T cycle uses the language command TCA (ToolChangeAbsolut) for the T call and therefore also has the possibility of inserting tools that have been disabled. However, this language command cannot be substituted.

The following must be observed if working at a machine with T replacement:

- The T preparation cycle must be explicitly called at the specified position.
- If the call is realized using the CUST_T cycle, then the parameter evaluation ($C_TS_PROG, …) is skipped.
  Alternatively, the original TCA language command can be reprogrammed.

Further information is available in the following document:

### References

Function Manual Basic Functions; Auxiliary function outputs to PLC (H2),
Section: "Behavior during block search" > "ASUB at the end of the SERUPRO"

## 18.3.5 CUST_TECHCYC.SPF manufacturer cycle

### Function

For technological cycles, the CUST_TECHCYC cycle is called from ShopMill and ShopTurn cycles, if the machine builder must make certain adaptations to a specific machine (e.g. activate specific M commands). For instance, this may be necessary in order to retract or extend part catchers when cutting off parts on lathes or to position a spindle for clamping (only in JobShop).

The CUST_TECHCYC.SPF cycle can be used to program and execute the necessary sequences.

## Adapt manufacturer cycle CUST_TECHCYC.SPF

You can execute the following actions with the aid of the _MODE parameter:

| Marker | Action | |
|---|---|---|
| _M1 | Main spindle: | Changeover to C axis mode |
| _M2 | | Changeover to spindle mode |
| _M3 | | Clamp C axis |
| _M4 | | Release C axis clamping |
| _M5 | | Rinse chuck |
| _M6 | | Close chuck |
| _M7 | | Open chuck when spindle is stationary |
| _M8 | | Open chuck when spindle is rotating |
| _M9 | | Flush chuck |
| | | |
| _M11 | Tool spindle: | Changeover to C axis mode |
| _M12 | | Changeover to spindle mode |
| _M13 | | Clamp C axis |
| _M14 | | Release C axis clamping |
| | | |
| _M21 | Counterspindle: | Changeover to C axis mode |
| _M22 | | Changeover to spindle mode |
| _M23 | | Clamp C axis |
| _M24 | | Release C axis clamping |
| _M25 | | Rinse chuck |
| _M26 | | Close chuck |
| _M27 | | Open chuck when spindle is stationary |
| _M28 | | Open chuck when spindle is rotating |
| _M29 | | Rinse chuck off |
| | | |
| _M30 | Position 4th axis after block search (e.g. counterspindle, tailstock, back rest) | |
| | | |
| _M41 | Engage driven tool | |
| _M42 | Disengage driven tool | |
| | | |
| _M61 | Changeover to turning | |
| _M62 | Changeover to peripheral surface machining C | |
| _M63 | Changeover to face machining C | |
| _M64 | Changeover to peripheral surface machining Y | |
| _M65 | Changeover to face machining Y | |
| _M66 | Changeover to face machining B (swivel) | |
| _M67 | Deselection of the face machining B (swivel) | |
| _M68 | After block search to ST block before swivel | |
| | | |

| Marker | Action | |
|---|---|---|
| _M100 | Drawer: | Position before cut-off |
| _M101 | | Open during cut-off |
| _M102 | | Close after cut-off |
| _M103 | Cut-off completed | |
| | | |
| _M110 | Tool change | |
| _M111 | After end of tool change cycle (with tool change) | |
| _M112 | Before tool change on retraction plane | |
| _M113 | Before cutting edge reselection | |
| _M114 | After end of tool change cycle (without tool change) | |
| _M115 | After tool preselection (ShopTurn) | |
| _M116 | At tool change point (ShopTurn) | |
| _M117 | Before tool change T command | |
| | | |
| _M120 | Definition: Coupling of counterspindle / main spindle | |
| _M121 | After COUPOF switch off position control | |
| _M122 | Definition: Coupling, main/counterspindle | |
| _M123 | After COUPOF switch off position control | |
| _M125 | Changeover to main spindle | |
| _M126 | Changeover to counterspindle | |
| | | |
| _M131 | Start of program (ShopTurn program) | |
| _M132 | End of program header (ShopTurn program) | |
| _M135 | End of program loop (ShopTurn program) | |
| _M136 | End of program (ShopTurn program) | |
| | | |
| _M140 | Block search | Before tool change |
| _M141 | | After tool change |
| _M142 | | After tool change (new tool from ShopTurn) |
| | | |
| _M150 | 2-channel stock removal: | Leading channel before Balance Cutting |
| _M151 | | Leading channel after Balance Cutting |
| _M152 | | Following channel before Balance Cutting |
| _M153 | | Following channel after Balance Cutting |
| _M154 | | Start machining in the leading channel |
| _M155 | | End machining in the leading channel |
| _M156 | | Start machining in the following channel |
| _M157 | | Start machining in the following channel |
| | | |
| _M160 | Start multichannel programming | |
| _M170 | Start block with spindle | |
| _M171 | End block with spindle | |
| | | |

| Marker | Action |
|---|---|
| _M231 | Start of program (ShopMill program) |
| _M232 | End of program header (ShopMill program) |
| _M235 | End of program loop (ShopMill program) |
| _M236 | End of program (ShopMill program) |
| | |
| _M300 | Boring CYCLE86<br>Adapt the spindle to the tool position (SD55216 $SCS_FUNCTION_MASK_DRILL_SET bit 6 = 1)<br>For specific machine kinematics, it may be necessary to make an additional adjustment of the correction angle for the spindle positioning (e.g. intermediate gear at the spindle). The _M300 mark is run through after the correction angle is calculated. There the correction angles _MEA_CORR_ANGLE[0] (for the manufacturer) and _MEA_CORR_ANGLE[1] (system) can be changed, if applicable. For subsequent spindle positioning operations, both angles act additively in CYCLE86. |

### 18.3.6 CUST_MULTICHAN user cycle

#### Function

The CUST_MULTICHAN cycle is used for multi-channel programming at lathes. It is called at the start of the block when programming with blocks.

In the cycle, e.g. the master spindle is set to the spindle specified at the start of the block. The return value (= 1) can be used to control whether the complete block is skipped.

#### CUST_MULTICHAN

| Parameter | Meaning |
|---|---|
| _S_NR | Spindle number to which the master spindle is set. |
| _RET | Return value |
| = 0 | The block is executed corresponding to the run-in mode. |
| = 1 | The entire block is skipped. |

## 18.4 Milling

### 18.4.1 General

All settings for machine and setting data of the standard cycles are also applicable for milling under ShopMill and turning under ShopTurn.

## 18.4.2 Technology cycles for milling

### Milling function screen

| SD52214 $SCS_function_MASK_MILL | Milling function screen |
|---|---|
| Bit 0 | Enable cylinder surface transformation (ShopMill) |
| Bit 1 | List table to accept blank (on milling machines)<br>(This bit only has to be set, if the machine has a rotary axis and a fixed table for accepting the blank.) |
| Bit 2 | reserved |
| Bit 3 | Enable machining inside/outside |
| Bit 4 | Enable spindle clamping (C axis) |
| Bit 5 | Enable spindle control of the tool spindle via user interface |
| Bit 6 | Enable spindle control of the main spindle via user interface |

| SD55214 $SCS_FUNCTION_MASK_MILL_SET | Milling function screen |
|---|---|
| Bit 0 | Basic setting, milling in climbing. |
| Bit 2 | Depth calculation of the milling cycles, with or without safety clearance. |
| = 0 | Depth calculation of the milling cycles is performed between the reference plane + safety clearance and the depth. |
| = 1 | Depth calculation is performed without including the safety clearance.<br>Bit 2 is effective in the following milling cycles: CYCLE61, CYCLE71, CYCLE76, CYCLE77, CYCLE79, CYCLE899, LONGHOLE, SLOT1, SLOT2, POCKET3, POCKET4. |

### Contour milling (CYCLE63, CYCLE64)

| SD55460 $SCS_MILL_CONT_INITIAL_RAD_FIN | Finishing approach circle radius<br>The radius of the approach circle during the finishing of contour pockets is affected. |
|---|---|
| = 0 | The radius is selected so that at the starting point the safety clearance to the finishing allowance is maintained (default value). |
| > 0 | The radius is selected so that at the starting point the value of this channel-specific setting data to the finishing allowance is maintained. |

| SD55212 $SCS_FUNCTION_MASK_TECH_SET | General function screen for all technologies |
|---|---|
| = 6 | |
| Bit 3 | Delete programs generated by contour cycles (CYCLE63, CYCLE64, CYCLE952) |
| = 0 | Generated programs are not deleted (compatibility as before) |
| = 1 | Generated programs are deleted as soon as they have been executed by the calling cycle. |

### Engraving (CYCLE60)

| SD55400 $SCS_MILL_ENGRAVE_POINT_RAD | Engraving cycle CYCLE60: Circular arc radius for generating the "dot" character |
|---|---|
| = 0 | |
| = 0 | The "dot" character is produced by a simple linear insertion movement of the tool (default setting for conventional engraving tools, compatibility). |
| > 0 | The "dot" character is executed as a circle with the radius value from this setting data (setting for some special tools). |

### Multiple edge (CYCLE79), circular position pattern (HOLES2), circumferential groove (SLOT2)

| SD55230 $SCS_CIRCLE_RAPID_FEED | Rapid feed in mm/min for positioning on a circular path between the circumferential grooves or the contour elements. |
|---|---|
| = 10000 | |

## 18.4.3 Setting-up ShopMill cycles for milling

### Load ShopMill cycles for milling

When the control boots, the ShopMill cycles are automatically loaded into the following directory: ../siemens/sinumerik/cycles/sm/prog

### Setting the coordinate system

| MD52000 $MCS_DISP_COORDINATE_SYSTEM | Position of the coordinate system |
|---|---|
| = 0 | |

**Examples:**

=0 for vertical milling machines

=16 for horizontal milling machines, boring mill

A detailed description of the MD52000 is provided in Section:

Activating turning/milling/drilling/grinding technologies (Page 349)

For plane selection when milling, appropriately set the following machine data:

| MD52005 $MCS_DISP_PLANE_MILL | Plane selection for milling |
|---|---|
| = 17 | |
| = 0 | In the cycle support |
| = 17 | G17 (default value) |
| = 18 | G18 |
| = 19 | G19 |

## Simulation and simultaneous recording

In order that simulation and simultaneous recording are displayed without any errors, set the machine data as described in the following section:

Setting the technology for simulation (Page 141)

## Defining the direction of rotation

In order that the direction of rotation is correctly displayed in the ShopMill user interface, and when programming ShopMill functions, the correct direction of rotation is executed, you must make some settings that are coordinated with one another. You must align these settings to the actual direction of rotation of the axis at the machine.

| MD52207 $MCS_USAGE_ATTRIB[n] | Attribute of the axes in the channel<br>[n] channel axis number |
|---|---|
| Bit 4 | Displayed direction of rotation for M3 is counter-clockwise (for spindles) |
| Bit 5 | Direction of rotation M3 corresponds to rotary axis minus (for spindles)<br>This bit must be set analog to PLC bit DB31, ... DBX17.6! |

## Specify other direction of rotation for position pattern

You can specify a position pattern for the tool axis. Rather than Bit 12, you must set Bit10 to active in the MD52207 so that, for example, the position pattern can be machined on rotary axis C with an appropriate swivel position.

| MD52207 $MCS_USAGE_ATTRIB[n] | Attribute of the axes in the channel<br>[n] channel axis number |
|---|---|
| Bit 2 | Rotates around 3rd geometry axis (for rotary axes) |
| Bit 4 | Displayed direction of rotation for M3 is counter-clockwise (for spindles)<br>The viewing direction can be selected. Either from the interior looking out or from outside looking toward the spindle.<br>However, the same selection must be made for all spindles. |
| Bit 5 | Direction of rotation for M3 corresponds to rotary axis minus (for spindles) |
| Bit 6 | Display rotary axis as offset target for measuring |
| Bit 7 | Offer rotary axis in position pattern<br>The orientation of the axis is determined with Bits 0 to 2.<br>If one of the Bits 10 to 12 is set, they determine the orientation of the axis for position patterns. |
| Bit 8 | Offer rotary axis for blank clamping (on milling machines)<br>The orientation of the axis is determined with Bits 0 to 2. |
| Bit 10 | Rotary axes turns around the 1st geometry axis (only for position pattern) |

## Additional channel-specific machine data

| MD52216 $MCS_FUNCTION_MASK_DRILL | Drilling function screen form |
|---|---|
| Bit 2 | Tapping also without encoder<br>The setting depends on whether at the machine the tool spindle is equipped with an encoder. |
| = 0 | Tapping only possible with encoder |

| | |
|---|---|
| = 1 | Tapping also possible without encoder |
| Bit 3 | Constant cutting speed referred to the diameter of the centering |
| = 0 | Constant cutting speed referred to the diameter of the tool |
| = 1 | Constant cutting speed referred to the diameter of the centering |

| MD52229 $MCS_ENABLE_QUICK_M_CODES | Enable fast M commands |
|---|---|
| = 0 | |
| Bit 0 | Coolant OFF (default value) |
| Bit 1 | Coolant 1 ON |
| Bit 2 | Coolant 2 ON |
| Bit 3 | Coolants 1 and 2 ON |

| MD52230 $MCS_M_CODE_ALL_COOLANTS_OFF | M code for all coolants Off |
|---|---|
| = 9 | You define the M function to switch off the coolant that is output when the tool is changed. |

| MD52231 $MCS_M_CODE_COOLANT_1_ON | M code for coolant 1 On |
|---|---|
| = 8 | You define the M function for coolant 1 that is output when the tool is changed. |

| MD52232 $MCS_M_CODE_COOLANT_2_ON | M code for coolant 2 On |
|---|---|
| = 7 | You define the M function for coolant 2 that is output when the tool is changed. |

| MD52233 $MCS_M_CODE_COOLANT_1_AND_2_ON | M code for both coolants On |
|---|---|
| = -1 | You define the M function for coolant 1 and 2 that is output when the tool is changed. |

| MD52281 $MCS_TOOL_MCODE_FUNC_ON[ ] | M code for tool-specific function ON |
|---|---|
| = -1 | M function is not output.<br>If both M commands of a function "= -1", the corresponding field is not displayed on the interface. |
| [0] | M code for tool-specific function 1 ON |
| [1] | M code for tool-specific function 2 ON |
| [2] | M code for tool-specific function 3 ON |
| [3] | M code for tool-specific function 4 ON |

| MD52282 $MCS_TOOL_MCODE_FUNC_OFF[ ] | M code for tool-specific function OFF |
|---|---|
| = -1 | M function is output.<br>If both M commands of a function "= -1", the corresponding field is displayed on the interface. |
| [0] | M code for tool-specific function 1 OFF |
| [1] | M code for tool-specific function 2 OFF |
| [2] | M code for tool-specific function 3 OFF |
| [3] | M code for tool-specific function 4 OFF |

### Channel-specific cycle setting data

| SD55212 $SCS_FUNCTION_MASK_TECH_SET | Cross-technology function screen |
|---|---|
| = 6 | |
| Bit 0 | Tool preselection active.<br>The next tool is prepared directly after a tool change.<br>Note: For a revolver, the setting data must be set to "0". |
| Bit 1 | Automatic calculation of the thread depth for metric threads. |
| Bit 2 | Take the thread diameter and thread depth from the table. |

## 18.4.4 Cylinder surface transformation (TRACYL)



**Software option**

You require the following software option in order to use this function:
"Transmit and peripheral surface transformation".

### Precondition

- There must be at least one rotary axis at the machine.
- The milling tool must be radially oriented to the cylinder to be machined.

### Function

The following groove machining operations can be performed with the Cylinder surface transformation functions:

- Longitudinal grooves on cylindrical bodies
- Transverse grooves on cylindrical objects
- Grooves with any path on cylindrical bodies

The path of the grooves is programmed with reference to the unwrapped, level surface of the cylinder. Programming can be realized using straight line/circle, drilling or milling cycles or contour milling (free contour programming).

There are two variants of cylinder surface transformation, i.e.

1. with groove side offset (ON)
2. without groove side offset (OFF)

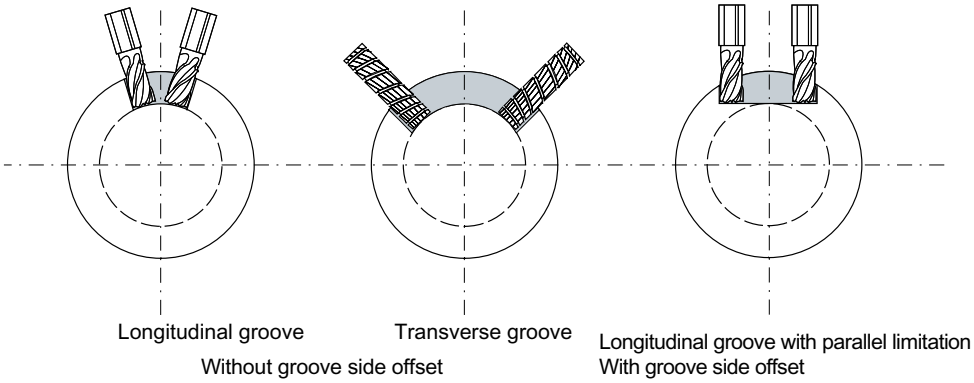


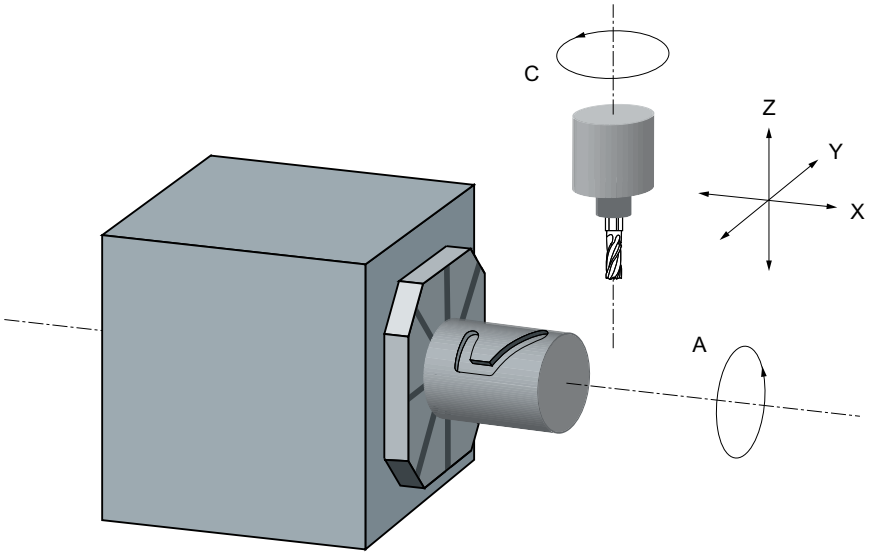

Figure 18-1 Grooves with and without groove side offset

## 18.4.5 Example: Milling machine with the XYZ-AC axis configuration

### Example: Milling machine with the XYZ-AC axis configuration

Example of an axis configuration on the following machine:

## Axis configuration




X 1st axis of the machining plane parallel to the rotary axis
Y 2nd axis of the machining plane
Z Infeed axis (tool axis) perpendicular (radial) to the rotary axis
A Rotary axis
C Working spindle

Figure 18-2 Machining slots on a cylinder surface with X-A-Z kinematics

You must configure two data sets with the following machine data for the machine illustrated above:

| MD20070 $MC_AXCONF_MACHAX_USED[4] | Machine axis number valid in channel |
|---|---|
| = 5 | Number of channel axes |

| MD20080 $MC_AXCONF_CHANAX_NAME_TAB[ ] | Name of channel axis in the channel |
|---|---|
| [0] = XC | Channel axis XC, corresponds to channel axis 1 |
| [1] = YC | Channel axis YC, corresponds to channel axis 2 |
| [2] = ZC | Channel axis ZC, corresponds to channel axis 3 |
| [3] = A | Channel axis A, corresponds to channel axis 4 |
| [4] = C | Channel axis C, corresponds to channel axis 5 |

| MD28082 $MC_MM_SYSTEMFRAME_MASK | System frames (SRAM) |
|---|---|
| **= 21H** | Default setting |
| Bit 0 = 1 | System frame for actual value setting and scratching |
| Bit 5 = 1 | System frame for cycles |
| Bit 6 = 1 | System frame for transformations |

## General settings for the transformation

| MD10602 $MN_FRAME_GEOAX_CHANGE_MODE | | Frames when switching over geometry screens |
|---|---|---|
| = 1 | The actual total frame (zero offsets) is recalculated when switching over geometry axes (selecting - deselecting TRACYL). | |

## Data set for the 1st transformation in the channel

| MD24100 $MC_TRAFO_TYPE_1 | | Definition of transformation 1 in the channel. |
|---|---|---|
| = 512 | Transformer type (512 = cylinder surface transformation without groove side offset) | |

| MD24110 $MC_TRAFO_AXES_IN_1 | | Axis assignment for the 1st transformation in the channel |
|---|---|---|
| [0] = 3 | Channel axis: Infeed axis (tool axis) perpendicular (radial) to the rotary axis | Z |
| [1] = 4 | Channel axis: Rotary axis | A |
| [2] = 1 | Channel axis: 1st axis of the machining plane parallel to the rotary axis | X |
| [3] = 2 | Channel axis: 2nd axis of the machining plane | Y |

| MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1 | | Assignment of the geometry axes to channel axes for transformation 1. |
|---|---|---|
| [0] = 1 | Channel axis: 1st geometry axis | X |
| [1] = 4 | Channel axis: 2nd geometry axis | A |
| [2] = 3 | Channel axis: 3rd geometry axis | Z |

| MD24800 $MC_TRACYL_ROT_AX_OFFSET_1 | |
|---|---|
| = 0 | Offset of rotary axis for the 1st TRACYL transformation. |

| MD24805 $MC_TRACYL_ROT_AX_FRAME_1 | |
|---|---|
| = 1 | Axial offset of rotary axis for the 1st TRACYL transformation. |

| MD24810 $MC_TRACYL_ROT_SIGN_IS_PLUS_1 | |
|---|---|
| = 1 | Sign of the rotary axis for the 1st TRACYL transformation. |

| MD24820 $MC_TRACYL_BASE_TOOL_1[ ] | |
|---|---|
| [0] = 0 | Vector of the base tool for the 1st TRANSMIT transformation in XYZ |
| [1] = 0 | |
| [2] = 0 | |

## Data set for the 2nd transformation in the channel

| MD24200 $MC_TRAFO_TYPE_2 | |
|---|---|
| = 513 | Transformer type (513 = cylinder surface transformation with groove side offset) |

| MD24210 $MC_TRAFO_AXES_IN_2 | | |
|---|---|---|
| [0] = 3 | Channel axis: Infeed axis perpendicular (radial) to rotary axis | Z |
| [1] = 4 | Channel axis: Rotary axis | A |
| [2] = 1 | Channel axis: 1st axis of the machining plane parallel to the rotary axis | X |
| [3] = 2 | Channel axis: 2nd axis of the machining plane | Y |

| MD24220 $MC_TRAFO_GEOAX_ASSIGN_TAB_2 | | |
|---|---|---|
| [0] = 1 | Channel axis: 1st geometry axis | X |
| [1] = 4 | Channel axis: 2nd geometry axis | A |
| [2] = 3 | Channel axis: 3rd geometry axis | Z |

| MD24850 $MC_TRACYL_ROT_AX_OFFSET_2 | |
|---|---|
| = 0 | Offset of the rotary axis for the 2nd TRACYL transformation |

| MD24855 $MC_TRACYL_ROT_SIGN_IS_FRAME_2 | |
|---|---|
| = 1 | Axial offset for the 2nd TRACYL transformation |

| MD24860 $MC_TRACYL_ROT_SIGN_IS_PLUS_2 | |
|---|---|
| = 1 | Sign of the rotary axis for the 2nd TRACYL transformation |

| MD24870 $MC_TRACYL_BASE_TOOL[ ] | |
|---|---|
| [0] = 0 | Vector of the base tool for the 2nd TRACYL transformation in XYZ |
| [1] = 0 | |
| [2] = 0 | |

**Note**

**Data sets for transformations**

For both data sets, you can use any transformations from all available transformations (MD24100 $MC_TRAFO_TYPE_1, MD24200 $MC_TRAFO_TYPE_2, etc.). The two data sets need not be directly next to each other.

However, the 1st data set must always be used for "Cylinder surface transformation **without** groove side offset" (= 512) and the 2nd data set for "Cylinder surface transformation **with** groove side offset" (=513).

## 18.4.6 ShopMill cycles for multiple clamping



**Software options**

You require the following software option in order to use this function: "ShopMill/ ShopTurn"

### Function

The Multiple clamping function provides optimized tool change over several workpiece clampings. On the one hand, this reduces downtimes and, on the other, eliminates tool change times because a tool performs as many machining operations as possible in all clampings before the next tool change is initiated.

You can either execute the same program several times for the clampings or you can select different programs. If you are using large-area fixture plates on your machine, you do not have to set up anything else.

In the case of rotating clamping devices, on the other hand, you must adapt a cycle to match the features of the clamping device so as to ensure that the next workpiece can be turned to the machining position after machining of the previous one (or for multiple clamping devices even while the current workpiece is being machined).

### Procedure for setting up

1. Change the CLAMP.SPF cycle which is stored in the cycle/sc/prog/templates_deu or cycle/sc/prog/templates_eng directory.
2. Copy the cycle into the user or manufacturer directory.

### Example

A rotating clamping device (reversible clamping device) is used with four clampings. The respective clampings can be machined by positioning rotary axis A:

| | |
|---|---|
| Clamping 1: | A = 0° |
| Clamping 2: | A = 90° |
| Clamping 3: | A = 180° |
| Clamping 4: | A = 270° |

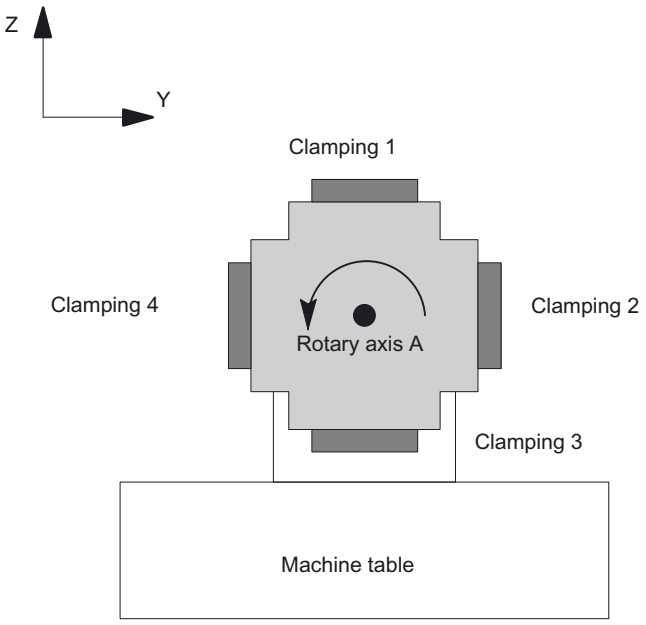



Figure 18-3 Rotating clamping device (reversible clamping device)

The cycle CLAMP.SPF must be adapted as follows:

```
...
DEF INT _NV ; Auxiliary variable
;-----------------------------------
;Adaptation
;
IF _ACT==1
G0 A=DC(0)
ENDIF
;
IF _ACT==2
G0 A=DC(90)
ENDIF
;
IF _ACT==3
G0 A=DC(180)
ENDIF
;
IF _ACT==4
G0 A=DC(270)
ENDIF
```

```
;
;-----------------------------------
_NV=_NPV+_ACT ;  Calculate current zero offset
N10 G[8]=_NV ;  There must not be a calculation here
RET
```

# 18.5 Turning

## 18.5.1 General

All settings for machine and setting data of the standard cycles are also applicable for milling under ShopMill and turning under ShopTurn.

## 18.5.2 Setting up cycles for turning

### Setting the coordinate system

| MD52000 $MCS_DISP_COORDINATE_SYSTEM | Position of the coordinate system |
|---|---|
| = 0 | Default setting |

Examples:

0: Vertical turning machine (carousel-type turning machine)
19: Horizontal turning machine, machining in front of the turning center
34:  Horizontal turning machine, machining behind the turning center (inclined bed turning machine)

A detailed description of MD52000 is given in Section: "Activating turning/milling/drilling/ grinding technologies (Page 349)."

### Turning function screen

Enable various functions under the Turning function screen in the following channel-specific configuration machine data.

| MD52218 $MCS_FUNCTION_MASK_TURN | Turning function screen |
|---|---|
| Bit 0 | Enable zoom under manual for tool measurement |
| Bit 1 | Enable part catcher for cut-off |
| Bit 2 | Enable tailstock |
| Bit 3 | Enable spindle control of main spindle via user interface |
| Bit 4 | Enable spindle control of the counterspindle control via user interface |
| Bit 5 | Enable spindle control of the tool spindle via user interface |

| | |
|---|---|
| Bit 6 | Enable balance cutting for two-channel stock removal |
| Bit 7 | Retraction when cutting along the contour with G1 |
| Bit 8 | Enter the spindle chuck data in the program |
| Bit 9 | Additional input of the tailstock data in the program |
| Bit 10 | Enable crowned thread |
| Bit 12 | Deactivate thread synchronization |
| Bit 13 | Cutting along the contour with CYCLE95 (828D programGUIDE without Advanced Technology |

## Meaning of the axes

You set the meaning of the axes in machine data MD52206 $MCS_AXIS_USAGE[n].

| MD52206 $MCS_AXIS_USAGE[n] | Meaning of the axes in the channel<br>[n] channel axis number |
|---|---|

A detailed description of MD52206 is provided in Chapter "Activating turning/milling/drilling/grinding technologies (Page 349)

## Traversing direction

In order that the user interface and cycles know in which direction the spindles and rotary axes rotate, the following settings must be made.

You set the direction of rotation of an NC rotary axis via MD32100 $MA_AX_MOTION_DIR.

| MD32100 $MA_AX_MOTION_DIR | Traversing direction |
|---|---|
| -1 | Direction reversal |
| 0, 1 | No direction reversal |

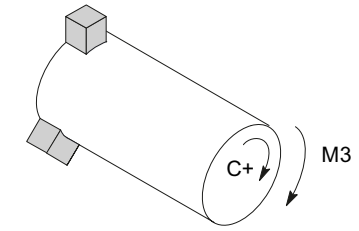
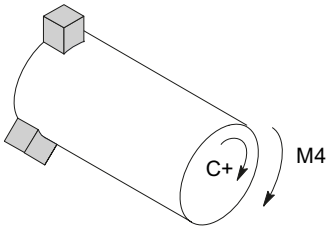
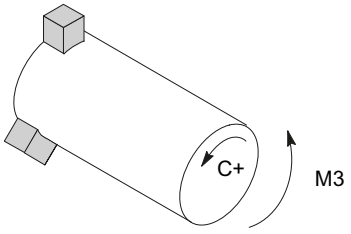
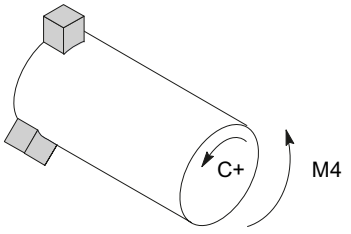
## Defining the direction of rotation

In order to ensure that the direction of rotation of the spindle and C axis is correctly displayed on the user interface and when programming, the correct direction of rotation is executed, you must make several settings that are coordinated with one another. You must base these settings on the actual direction of rotation of the spindle/C axis on the machine.

You specify the settings in the following machine data:

| MD52207 $MCS_AXIS_USAGE_ATTRIB[n] | Attributes of the axes<br>[n] channel axis number |
|---|---|
| Bit 3 | Displayed positive direction of rotation is counter-clockwise (for rotary axes)<br>The view must be from the inside of the machine (operator's view) to the axis of rotation. |
| Bit 4 | Displayed direction of rotation for M3 is counter-clockwise (for spindles)<br>This bit can be set from the operator's view or according to DIN 66025. |
| Bit 5 | Direction of rotation for M3 corresponds to minus rotary axis (for spindles)<br>This bit must be set analogous to PLC bit DB31, ... DBX17.6! |

As a result, the following setting options are obtained for the main spindle:

| Main spindle direction of rotation | 52207[n] bit 3 = | 52207[n] bit 4 = *) | 52207[n] bit 5 = | DB31, ... DBX17.6 |
|---|---|---|---|---|
|  | 0 | 0 | 0 | 0 |
|  | 0 | 1 | 1 | 1 |
|  | 1 | 1 | 0 | 0 |
|  | 1 | 0 | 1 | 1 |

**Note**

**MD52207[n] bit 4 = *)**

Bit 4 can be set from the operator's view (values as specified) or according to DIN ("0" and "1" are then interchanged).

## Dimensions

To define the reference point for moving the counterspindle, you must first communicate the dimensions of the counterspindle. You can either enter the dimensions in the following axis-specific cycle machine data or in menu "Parameter" → "Setting data" → "Spindle chuck data". Changes to the machine data are automatically accepted in the menu and vice versa.

| MD53240 $MAS_SPINDLE_PARAMETER[ ] | | Spindle chuck data |
|---|---|---|
| [0] | Chuck dimensions | |

| | |
|---|---|
| [1] | Stop dimensions |
| [2] | Jaw dimensions |

| MD53241 $MAS_SPINDLE_CHUCK_TYPE[ ] | Spindle jaw type |
|---|---|
| = 0 | Clamping, outer |
| = 1 | Clamping, inner |

## Additional settings

The M code, e.g. M34 or M1 = 34 for the spindle chuck is defined in the following machine data. The manufacturer cycle CUST_TECHCYC.SPF also takes the M functions from the following machine data:

| MD52250 $MCS_M_CODE_CHUCK_OPEN[ ] | M code for open chuck with stationary spindle |
|---|---|
| [0] | Main spindle |
| [1] | Counterspindle |

| MD52251 $MCS_M_CODE_CHUCK_CLOSE_OPEN_ROT[ ] | M code for open chuck with spindle rotating |
|---|---|
| [0] | Main spindle |
| [1] | Counterspindle |

| MD52252 $MCS_M_CODE_CHUCK_CLOSE[ ] | M code for close chuck |
|---|---|
| [0] | Main spindle |
| [1] | Counterspindle |

| MD52253 $MCS_M_CODE_TAILSTOCK_FORWARD[ ] | M code for quill forward |
|---|---|
| [0] | Quill opposite the main spindle |
| [1] | |

| MD52254 $MCS_M_TAILSTOCK_BACKWARD[ ] | M code for quill back |
|---|---|
| [0] | Quill opposite the main spindle |
| [1] | Quill opposite the main spindle |

### Contour turning (CYCLE952)

| SD55212 $SCS_FUNCTION_MASK_TECH_SET | General function screen for all technologies |
|---|---|
| = 6 | |
| Bit 3 | Delete programs generated by contour cycles (CYCLE63, CYCLE64, CYCLE952) |
| = 0 | Generated programs are not deleted (compatibility as before) |
| = 1 | Generated programs are deleted as soon as they have been executed by the calling cycle. |

### Rounding the contour

| SD55582 $SCS_TURN_CONT_TRACE_ANGLE | Contour turning: Minimum angle for rounding the contour |
|---|---|
| = 5 | Specifies the angle between the cutting edge and contour, above which for contour turning, the contour is rounded in order to remove residual material (default value). |

| SD 55218 $SCCS_FUNCTION_MASK_TURN_SET | Turning function screen |
|---|---|
| Bit 3 | Chamfer in CYCLE930 as chamfer length (CHF) |
| Bit 4 | Retraction from internal machining (CYCLE951) |
| = 0 | Compatibility as before<br>If, during the cycle, the necessity for tightening is detected at the corner, the tool is retracted from the internal machining, otherwise it is not. |
| = 1 | Retraction from internal machining is generally in the cutting axis. |

## 18.5.3 Setting-up ShopTurn cycles for turning

| MD52210 $MCS_FUNCTION_MASK_DISP | Function screen display |
|---|---|
| Bit 0 = 1 | Dimension system for programs always in the basic system (default setting) |
| Bit 1 = 1 | Face view when turning (default setting) |
| Bit 4 = 1 | Display follow-on tool in the T,F,S window |

| MD52212 $MCS_FUNCTION_MASK_TECH | Cross-technology function screen |
|---|---|
| Bit 2 | Approach logic for stepped drill |
| Bit 3 | Block search cycle for ShopMill / ShopTurn |
| = 0 | The E__ASUP or F_S_ASUP cycles are not called in the block search cycle PROG_EVENT.SPF |
| = 1 | The E__ASUP (under ShopMill) or F_S_ASUP (under ShopTurn) cycles are called in the block search cycle PROG_EVENT.SPF |
| Bit 4 | Approach logic using the cycle |
| Bit 5 | The cycle for SERUPRO (CYCLE207) is called in the block search cycle PROG_EVENT.SPF.<br>Bit 3 has no effect when bit 5 is set. |
| Bit 6 | Work offset value WO cannot be entered |
| Bit 7 | Identification of the end of tool lives in the program (ShopMill / ShopTurn) |
| Bit 8 | Manual machine (ShopMill/ShopTurn) |
| Bit 9 | Selection/deselection of the work offset using softkey |
| Bit 11 | Deactivate position check for drilling and milling tools |

| MD52214 $MCS_FUNCTION_MASK_MILL | Milling function screen |
|---|---|
| Bit 3 | "Inner/rear" machining is enabled in the ShopTurn screens which themselves define the machining plane. |

| | |
|---|---|
| Bit 4 | If you have implemented the "Clamp/release spindle" function using the manufacturer cycle CUST_TECHCYC.SPF, then using this machine data, you can activate the "Clamp/ release spindle" parameter in the drilling and milling screens. |
| = 0 | The "Clamp/release spindle" parameter is not displayed in the drilling and milling screens. ShopTurn automatically clamps the spindle if it makes sense for the particular machining operation. |
| = 1 | The "Clamp/release spindle" parameter is displayed in the drilling and milling screens. You decide for which machining operation the spindle should be clamped. |

| MD52216 $MCS_FUNCTION_MASK_DRILL | Drilling function screen |
|---|---|
| Bit 3 | Constant cutting speed referred to the diameter of the centering |
| = 0 | Constant cutting speed referred to the diameter of the tool |
| = 1 | Constant cutting speed referred to the diameter of the centering |

| MD52218 $MCS_FUNCTION_MASK_TURN | Turning function screen |
|---|---|
| Bit 2 | Enable tailstock |
| Bit 5 | Enable spindle control of the tool spindle via user interface |
| Bit 6 | Enable Balance Cutting for two-channel stock removal |

| MD52229 $MCS_ENABLE_QUICK_M_CODES | Enable fast M commands |
|---|---|
| = 0 | |
| Bit 0 | Coolant OFF |
| Bit 1 | Coolant 1 ON |
| Bit 2 | Coolant 2 ON |
| Bit 3 | Coolants 1 and 2 ON |

| MD52230 $MCS_M_CODE_ALL_COOLANTS_OFF | M code for all coolants off |
|---|---|
| = 9 | This machine data is used to define the M function for switching off the coolant; which is output when the tool is changed. |

| MD52231 $MCS_M_CODE_COOLANT_1_ON | M code for coolant 1 on |
|---|---|
| = 8 | This machine data is used to define the M function for coolant 1, which is output when the tool is changed. |

| MD52232 $MCS_M_CODE_COOLANT_2_ON | M code for coolant 2 on |
|---|---|
| = 7 | This machine data is used to define the M function for coolant 2, which is output when the tool is changed. |

| MD52233 $MCS_M_CODE_COOLANT_1_AND_2_ON | M code for both coolants on |
|---|---|
| = -1 | This machine data is used to define the M function for coolant 1 and 2, which is output when the tool is changed. |

| SD55505 $SCS_TURN_ROUGH_O_RELEASE_DIST | Retraction distance for stock removal during external machining |
|---|---|
| = 1 | Specifies the distance, by which the tool is retracted from the contour when removing stock from an outer corner. This does not apply to stock removal at a contour (default value). |
| = -1 | The distance is internally defined. |

| SD55506 $SCS_TURN_ROUGH_'I_RELEASE_DIST | Retraction distance for stock removal during internal machining |
|---|---|
| = 0.5 | Specifies the distance, by which the tool is retracted from the contour when removing stock from an internal corner. This does not apply to stock removal at a contour (default value). |
| = -1 | The distance is internally defined. |

## 18.5.4 Setting up a counterspindle

**Software option**

You require the following software option in order to use the counterspindle on your machine:

"Generic coupling 'CP-STATIC'" or higher

### Function

If your turning machine has a counterspindle, you can machine workpieces using turning, drilling and milling functions on the front and rear faces without reclamping the workpiece manually.

Before machining the rear face, the counterspindle must grip the workpiece, pull it out of the main spindle, and position it at the new machining position.

### Meaning of the axes

| MD52206 $MCS_AXIS_USAGE[n] | Meaning of the axes in the channel<br>[n] channel axis number |
|---|---|

A detailed description of MD52006 is provided in Chapter "Activating turning/milling/drilling/grinding technologies (Page 349)".

### Traversing direction

In order that the user interface and cycles know in which direction the spindles and rotary axes rotate, the following settings must be made.

You set the direction of rotation of an NC rotary axis via MD32100 $MA_AX_MOTION_DIR.

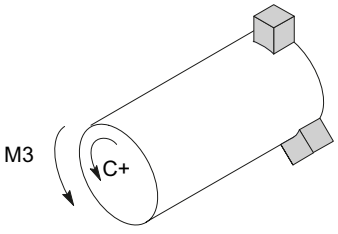
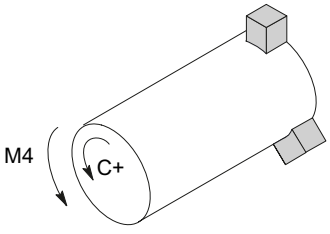
| MD32100 $MA_AX_MOTION_DIR | Traversing direction (not control direction) |
|---|---|
| -1 | Direction reversal |
| 0, 1 | No direction reversal |

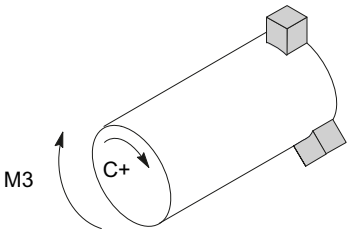
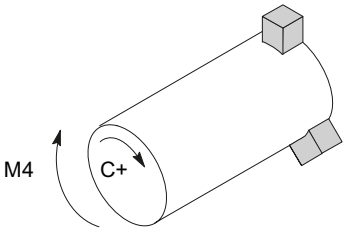
## Defining the direction of rotation

You set the directions of rotation for the user interface in the following machine data:

| MD52207 $MCS_AXIS_USAGE_ATTRIB[n] | Attributes of the axes in the channel<br>[n] channel axis number |
|---|---|
| Bit 3 | Displayed positive direction of rotation is counter-clockwise (for rotary axes)<br>The view must be from the inside of the machine (operator's view) to the axis of rotation. |
| Bit 4 | Displayed direction of rotation for M3 is counter-clockwise (for spindles)<br>This bit can be set from the operator's view or according to DIN. |
| Bit 5 | Direction of rotation for M3 corresponds to minus rotary axis (for spindles)<br>This bit must be set analogous to PLC bit DB31, ... DBX17.6! |

As a result, the following setting options are obtained for the counterspindle:

| **Counterspindle direction of rotation** | **52207[n] bit 3 =** | **52207[n] bit 4 = *)** | **52207[n] bit 5 =** | **DB31, ... DBX17.6** |
|---|---|---|---|---|
|  | 1 | 1 | 0 | 0 |
|  | 1 | 0 | 1 | 1 |

| Counterspindle direction of rotation | 52207[n] bit 3 = | 52207[n] bit 4 = *) | 52207[n] bit 5 = | DB31, ... DBX17.6 |
|---|---|---|---|---|
|  | 0 | 0 | 0 | 0 |
|  | 0 | 1 | 1 | 1 |

**Note**

**MD52207[n] bit 4 = *)**

Bit 4 can be set from the operator's view (values as specified) or according to DIN ("0" and "1" are then interchanged).

The display must be the same for the main spindle and the counterspindle ("from operator's view" or "according to DIN").

## Dimensions

To define the reference point for moving the counterspindle, you must first announce the dimensions of the counterspindle. You can either enter the dimensions in the following axis-specific cycle machine data or in menu "Parameter" → "Setting data" → "Spindle chuck data". Changes to the machine data are automatically accepted in the menu and vice versa.

| MD53240 $MAS_SPINDLE_PARAMETER[ ] | Spindle chuck data |
|---|---|
| [0] | Chuck dimensions |
| [1] | Stop dimensions |
| [2] | Jaw dimensions |

| MD52241 $MCS_SPINDLE_CHUCK_TYPE | Spindle jaw type |
|---|---|
| = 0 | Clamping, outer |
| = 1 | Clamping, inner |

## 18.5.5 Setting up the counterspindle under ShopTurn



**Software option**

You require the following software option in order to use the counterspindle on your machine:

"Generic coupling 'CP-STATIC'" or higher

The position to which the counterspindle travels when the program starts is defined in the following channel-specific cycle setting data:

| | |
|---|---|
| SD55232 $SCS_SUB_SPINDLE_REL_POS | Retraction position Z for counterspindle |

The following channel-specific setting data becomes effective when traveling to the fixed stop:

| | |
|---|---|
| SD55550 $SCS_TURN_FIXED_STOP_DIST | Distance for travel to fixed stop. |
| SD55551 $SCS_TURN_FIXED_STOP_FEED | Feedrate for travel to fixed stop. |
| SD55552 $SCS_TURN_FIXED_STOP_FORCE | Force for travel to fixed stop in %. |

Between traveling to the fixed stop and gripping, the counterspindle can retract a short distance to counteract compressive stress in the workpiece.

| | |
|---|---|
| SD55553 $SCS_TURN_FIXED_STOP_RETRACTION | Retraction distance before clamping after fixed stop. |

After gripping you can cut off the workpiece. Before doing so, the counterspindle can retract a short distance with the workpiece to exert tensile stress on the workpiece. This relieves pressure on the tool when cutting off.

| | |
|---|---|
| SD55543 $SCS_TURN_PART_OFF_RETRACTION | Retraction distance before cut-off. |

After cut-off, you can carry out a cut-off check and for turning, use the "Travel to fixed stop" function. You can activate/deactivate the cut-off check using the channel-specific cycle setting data:

| | |
|---|---|
| SD55540 $SCS_TURN_PART_OFF_CTRL_DIST | Distance for cut-off check. |
| SD55541 $SCS_TURN_PART_OFF_CTRL_FEED | Feedrate for cut-off check. |
| SD55542 $SCS_TURN_PART_OFF_CTRL_FORCE | Force for cut-off check in %. |

The cut-off is successful when travel to fixed stop fails. The following alarms are output:

| Alarm | Alarm text |
|---|---|
| 20091 | Axis %1 has not reached the fixed stop. |
| 20094 | Axis %1 end stop has been aborted. |

You can switch off the alarm display using the following machine data:

| | |
|---|---|
| MD37050 $MA_FIXED_STOP_ALARM_MASK | Enabling the fixed stop alarms. |
| = 2 | Suppressing alarms 20091 and 20094. |

You can set this machine data axis-specifically in the "Machine Data" window in the "Tool zero" operating area.

If, however, the specified force is reached during the cut-off check (i.e. travel to fixed stop is successful), alarm 61255 "Error during cut-off: Tool break?" is issued.

**Note**

The "Travel to fixed stop" function can also be used when gripping the spindle (see above). If travel to fixed stop does not succeed when gripping, an alarm will of course still be issued. Instead of alarms 20091 and 20094, the alarm 61254 "Error during travel to fixed stop" will be issued.

**Note**

MD37050 can also contain values not equal to 2, e.g. 10 = 2 + 8. Make sure that the value 2 is always contained logically so that the monitoring for travel to fixed stop can be hidden. If MD37050 does not contain the value 2, the cut-off check is not active.

### Machine manufacturer cycle

If you perform one of the following actions, you must adapt the machine manufacturer cycle CUST_TECHCYC.SPF.

- Switch between spindle and C axis mode for the main spindle or counterspindle.
- Open, close, flush chuck (main spindle / counterspindle).
- Change default settings for the coupling of the main spindle and counterspindle.

Please refer to the following section for the procedure:

Setting-up ShopTurn cycles for turning (Page 379)

## 18.5.6 Technology cycles for turning

### Thread-cutting (CYCLE99)

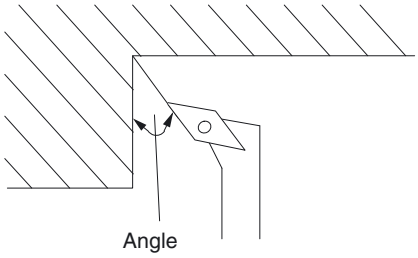
To correctly machine the thread, it is necessary to set the machine data MD52207 $MCS_AXIS_USAGE_ATTRIB[n] with bit 3, see Chapter: Setting up a counterspindle (Page 381).

### Contour groovingCYCLE930, contour turning CYCLE950, stock removal at corner CYCLE951, contour turning CYCLE952

| SD55500 $SCS_TURN_FIN_FEED_PERCENT | | Enter the finishing feedrate for complete machining, roughing and finishing. The percentage of the value corresponds to that entered under parameter F (feedrate). |
|---|---|---|
| = 100 | 100% finishing feedrate | |

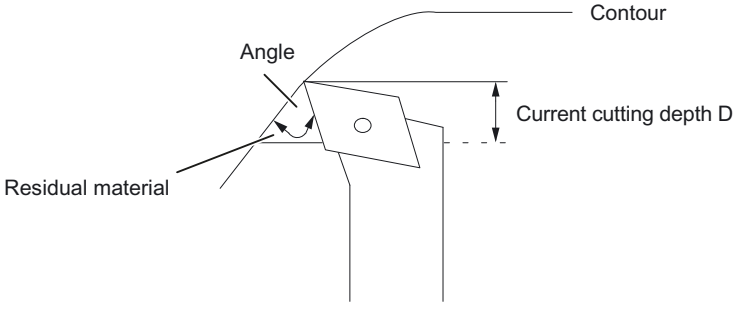
| SD55510 $SCS_TURN_GROOVE_DWELL_TIME | | Dwell time, which is necessary between grooving and retracting for grooving technology. Tool clearance time during grooving at the base. |
|---|---|---|
| = > 0 | Dwell time in seconds | |
| = < 0 | Dwell time in spindle revolutions | |

| SD55580 $SCS_TURN_CONT_RELEASE_ANGLE | | Angle, through which the tool is lifted from the contour for contour turning, roughing. |
|---|---|---|
| = 45 | Retraction angle of 45 degrees | |




| SD55581 $SCS_TURN_CONT_RELEASE_DIST | | Amount, by which the tool is lifted when roughing a contour taking into account the retraction angle SD55580 $SCS_TURN_CONT_RELEASE_ANGLE.<br>This also applies to stock removal, grooving and plunge turning. |
|---|---|---|
| = 1 | 1 mm or 1 inch retraction distance | |

| SD55582 $SCS_TURN_CONT_TRACE_ANGLE | | The angle between cutting edge and contour as of which rounding is performed on the contour during contour turning in order to remove residual material.<br>If the angle of the residual material is greater than that specified in the setting data, the tool will round the contour. |
|---|---|---|
| = 5 | 5 degree angle | |



| SD55583 $SCS_TURN_CONT_VARIABLE_DEPTH | | Percentage of the changing cutting depth when turning a contour<br>You can select changing the cutting depth for stock removal and for removing residual material. |
|---|---|---|
| = 20 | 20% variable cutting depth | |

| SD55584 $SCS_TURN_CONT_BLANK_OFFSET | | Safety clearance when avoiding obstacles, to avoid collisions during approach or retraction from the machining.<br>This also applies to stock removal, grooving and plunge turning. |
|---|---|---|
| = 1 | 1 mm or 1 inch blank allowance | |

| SD55585 $SCS_TURN_CONT_INTERRUPT_TIME | | Time for the feedrate interruption for contour turning.<br>This also applies to stock removal, grooving and plunge turning. |
|---|---|---|
| = > 0 | Interruption time in seconds | |
| = < 0 | Interruption time in revolutions | |
| = 0 | No interruption | |

**Note**

The channel-specific cycle setting data SD55585 is only evaluated if SD55586 $SCS_TURN_CONT_INTER_RETRACTION = 0.

| SD55586 $SCS_TURN_CONT_INTER_RETRACTION | | Retraction distance for contour turning for feedrate interruption.<br>This also applies to stock removal, grooving and plunge turning. |
|---|---|---|
| = > 0 | Retraction distance at feed interruption<br>SD55585 $SCS_TURN_CONT_INTERRUPT_TIME has no effect. | |
| = 0 | No retraction distance | |

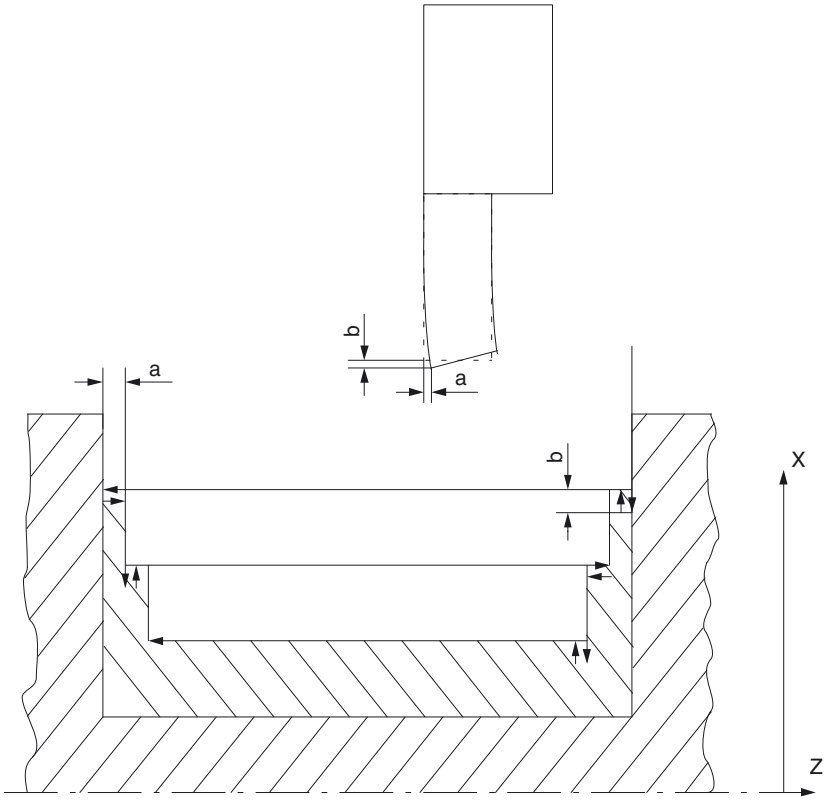
| SD55587 $SCS_TURN_CONT_MIN_REST_MAT_AX1 | | Limit value during contour turning for removal of residual material in the direction of axis 1 (for G18 Z).<br>This also applies to residual machining – stock removal, grooving and plunge turning. |
|---|---|---|
| 50 | The roughing step for residual machining is not executed, if the material to be removed is less than 50% of the finishing allowance. The material is only removed when finishing. | |

Example: If the SD is set to 50% and the final machining allowance is 0.5 mm, any residual material less than 0.25 mm is not removed with the residual machining – but is removed during finishing. If during a machining step, less residual material is present than defined in the SD, the error message "No material present" is issued. This means that this residual material roughing step can be omitted because no machining is performed.

| SD55588 $SCS_TURN_CONT_MIN_REST_MAT_AX2 | | Limit value during contour turning for removal of residual material in the direction of axis 2 (for G18 X).<br>This also applies to residual machining – stock removal, grooving and plunge turning. |
|---|---|---|
| 50 | The roughing step for residual machining is not executed, if the material to be removed is less than 50% of the finishing allowance. The material is only removed when finishing. | |

As the tool bends during plunge turning, the tool cannot travel right up to the contour during stock removal. The lateral distance to the last cut by which the next cut is shortened is specified in the following channel-specific cycle setting data.

| SD55595 $SCS_TURN_CONT_TOOL_BEND_RETR | | Retraction distance because of tool bending |
|---|---|---|
| 0.1 | 0.1 mm or 0.1 inch retraction distance | |




a Distance to the last cut, SD55595

b Retraction between plunge-cutting and stock removal, SD55596

As the tool bends during plunge turning, the tool would make an excessively deep cut during stock removal. The retraction distance of the tool between plunge-cutting and stock removal is specified in the following channel-specific cycle setting data:

| SD55596 $SCS_TURN_CONT_TURN_RETRACTION | | Retraction depth before turning |
|---|---|---|
| = 0.1 | 0.1 mm or 0.1 inch retraction depth | |

## 18.5.7 Axis configuration of a lathe

### General configuration

If driven milling tools are available on a lathe, then the following functions can also be set-up on this machine:

- Cylinder surface transformation (TRACYL) (Page 390)
- End face machining (TRANSMIT) (Page 394)

### Lathe with X and Z axes, main and tool spindle

For example, for a lathe with X and Z axes, main spindle (C1) and tool spindle (WZ), you can configure the following channel-specific machine data:

| MD20080 $MC_AXCONF_CHANAX_NAME_TAB[ ] | Channel axis name in the channel |
|---|---|
| [0] = XC | Channel axis XC |
| [1] = ZC | Channel axis ZC |
| [2] = C1 | Main spindle C1 |
| [3] = WZ | Tool spindle WZ |

### Lathe with X and Z axes, main and tool spindle and counterspindle

For example, for a lathe with X and Z axes, main spindle (C1), tool spindle (WZ) and counterspindle (C2), you can configure the following machine data:

| MD20080 $MC_AXCONF_CHANAX_NAME_TAB[ ] | Channel axis name in the channel |
|---|---|
| [0] = XC | Channel axis XC |
| [1] = ZC | Channel axis ZC |
| [2] = C1 | Main spindle C1 |
| [3] = WZ | Tool spindle WZ |
| [5] = C2 | Counterspindle C2 |

### Lathe with X and Z axes, main and tool spindle and Y axis

For example, for a lathe with X, Z and Y axes, main spindle (C1) and tool spindle (WZ), you can configure the following machine data:

| MD20080 $MC_AXCONF_CHANAX_NAME_TAB[ ] | Channel axis name in the channel |
|---|---|
| [0] = XC | Channel axis XC |
| [1] = ZC | Channel axis ZC |
| [2] = C1 | Main spindle C1 |
| [3] = WZ | Tool spindle WZ |
| [5] = YC | Channel axis YC |

## 18.5.8 Cylinder surface transformation (TRACYL)



**Software option**

You require the following software option in order to use this function:

"Transmit and peripheral surface transformation"

### Function

Using the cylinder surface transformation function (TRACYL), you can machine the peripheral surface of a turned part.

## Setting up

General settings for cylinder transformation

| MD10602 $MN_FRAME_GEOAX_CHANGE_MODE | Frames when switching over geometry axes. |
|---|---|
| = 1 | The actual total frame (work offsets) and when switching over geometry axes (selecting - deselecting TRACYL) is recalculated. |

| MD24040 $MC_FRAME_ADAPT_MODE | Adapting the active frames. |
|---|---|
| Bit 0 = 1 | Rotations in active frames that rotate coordinate axes for which there are no geometry axes, are deleted from the active frames. |
| Bit 1 = 1 | Shearing angles in active frames are made orthogonal. |
| Bit 2 = 1 | Scaling factors of all geometry axes in the active frames are set to 1. |

| MD28082 $MC_MM_SYSTEM_FRAME_MASK | |
|---|---|
| Bit 6 = 1 | Configuration of the transformation frame for the compensation. |

When setting up the functions, you can take the following channel-specific machine data into account:

| MD24300 $MC_TRAFO_TYPE_3 | Cylinder surface transformation, main spindle: Transformation 3 |
|---|---|
| = 512 | Without groove side offset (without Y axis) |
| = 513 | With groove side offset (with Y axis) |
| = 514 | With or without groove side offset and Y offset |

| MD24400 $MC_TRAFO_TYPE_4 | Cylinder surface transformation, counterspindle: Transformation 4 |
|---|---|
| = 512 | Without groove side offset (without Y axis) |
| = 513 | With groove side offset (with Y axis) |
| = 514 | With or without groove side offset and Y offset |

**Note**

You must also set up other machine data for each of the individual transformations.

## Cylinder surface transformation without groove side offset

| MD24300 $MC_TRAFO_TYPE_3 | Definition of the 3rd transformation in the channel. |
|---|---|
| = 512 | TRACYL main spindle. |

| MD24310 $MC_TRAFO_AXES_IN_3[ ] | Axis assignment for transformation 3. |
|---|---|
| [0] = 1 | Perpendicular to the rotary axis XC |

| | |
|---|---|
| [1] = 3 | Rotary axis (main spindle) C1 |
| [2] = 2 | Parallel to the rotary axis ZC |

| MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[ ] | Assignment of geometry axes to channel axes for transformation 3. |
|---|---|
| [0] = 1 | 1st channel axis X |
| [1] = 3 | 2nd channel axis Y |
| [2] = 2 | 3rd channel axis Z |

| MD24800 $MC_TRACYL_ROT_AX_OFFSET_1 | Offset of the rotary axis for the 1st TRACYL transformation. |
|---|---|
| = 0 | |

| MD24805 $MC_TRACYL_ROT_AX_FRAME_1 | Axial offset of the rotary axis is taken into account during TRACYL. |
|---|---|
| = 2 | |

| MD24810 $MC_TRACYL_ROT_SIGN_IS_PLUS_1 | Sign of the rotary axis for the 1st TRACYL transformation. |
|---|---|
| = 1 | |

| MD24820 $MC_TRACYL_BASE_TOOL_1[ ] | Vector of the base tool for the 1st TRACYL transformation. |
|---|---|
| [0] = 0 | |
| [1] = 0 | |
| [2] = 0 | |

## Setting up cylinder surface transformation under ShopTurn

In addition, set the following channel-specific configuration machine data:

| MD52214 $MCS_FUNCTION_MASK_MILL | Milling function screens |
|---|---|
| Bit 3 | Enable "internal / rear" machining in ShopTurn screens that define the machining plane themselves. |
| Bit 4 | If you have implemented the "Clamp/release spindle" function using the machine manufacturer cycle CUST_TECHCYC.SPF, then using this machine data, you can activate the "Clamp/release spindle" parameter in the drilling and milling screens. |
| = 0 | The "Clamp/release spindle" parameter is not displayed in the drilling and milling screens.<br>ShopTurn automatically clamps the spindle if it makes sense for the particular machining operation. |
| = 1 | The "Clamp/release spindle" parameter is displayed in the drilling and milling screens.<br>The operator decides for which machining operation the spindle should be clamped. |

## Cylinder surface transformation with groove side offset

| MD24300 $MC_TRAFO_TYPE_3 | | Definition of the 3rd transformation in the channel. |
|---|---|---|
| = 513 | TRACYL main spindle | |

| MD24310 $MC_TRAFO_AXES_IN_3[ ] | | Axis assignment for transformation 3. |
|---|---|---|
| [0] = 1 | Perpendicular to the rotary axis XC | |
| [1] = 3 | Rotary axis (main spindle) C1 | |
| [2] = 2 | Parallel to the rotary axis ZC | |
| [3] = 6 | Parallel to the cylinder surface and perpendicular to the rotary axis ZC | |

| MD24320 $MC_TRAFO_GEOAX_ASSIGN_TAB_3[ ] | | Assignment of geometry axes to channel axes for transformation 3. |
|---|---|---|
| [0] = 1 | 1st channel axis X | |
| [1] = 3 | 2nd channel axis Y | |
| [2] = 2 | 3rd channel axis Z | |

| MD24800 $MC_TRACYL_ROT_AX_OFFSET_1 | | Offset of the rotary axis for the 1st TRACYL transformation. |
|---|---|---|
| = 0 | | |

| MD24805 $MC_TRACYL_ROT_AX_FRAME_1 | | Axial offset of the rotary axis is taken into account during TRACYL. |
|---|---|---|
| = 2 | | |

| MD24810 $MC_TRACYL_ROT_SIGN_IS_PLUS_1 | | Sign of the rotary axis for the 1st TRACYL transformation. |
|---|---|---|
| = 1 | | |

| MD24820 $MC_TRACYL_BASE_TOOL_1[ ] | | Vector of the base tool for the 1st TRACYL transformation. |
|---|---|---|
| [0] = 0 | | |
| [1] = 0 | | |
| [2] = 0 | | |

## 18.5.9 End face machining (TRANSMIT)



**Software option**

You require the following software option in order to use this function:

"Transmit and peripheral surface transformation"

### Function

Using the end face machining function (TRANSMIT), you can machine the end face of a turned part.

### Setting up

For general settings for transformations, please refer to Chapter "Cylinder surface transformation (TRACYL) (Page 390)".

You can make additional settings in the following channel-specific machine data:

| MD24100 $MC_TRAFO_TYPE_1 | End face machining, main spindle: Transformation 1 |
|---|---|
| = 256 | Machining without Y axis |
| = 257 | Machining with Y axis |

| MD24200 $MC_TRAFO_TYPE_2 | End face machining, counterspindle: Transformation 2 |
|---|---|
| = 256 | Machining without Y axis |
| = 257 | Machining with Y axis |

| MD24110 $MC_TRAFO_AXES_IN_1[ ] | Axis assignment for the 1st transformation in the channel |
|---|---|
| [0] = 1 | Perpendicular to the rotary axis XC |
| [1] = 3 | Rotary axis (main spindle) C1 |
| [2] = 2 | Parallel to the rotary axis ZC |

| MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[ ] | Assignment of the geometry axes to channel axes for transformation 1 |
|---|---|
| [0] = 1 | 1st channel axis X |
| [1] = 3 | 2nd channel axis Y |
| [2] = 2 | 3rd channel axis Z |

| MD24900 $MC_TRANSMIT_ROT_AX_OFFSET_1 | |
|---|---|
| = 0 | Offset of the rotary axis for the 1st TRANSMIT transformation. |

| MD24905 $MC_TRANSMIT_ROT_AX_FRAME_1 | |
|---|---|
| = 2 | Axial offset of the rotary axis is taken into account during TRANSMIT 1. |

| MD24910 $MC_TRANSMIT_ROT_SIGN_IS_PLUS_1 | |
|---|---|
| = 0 | Sign of the rotary axis for the 1st TRANSMIT transformation. |

| MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1 | |
|---|---|
| = 1 | Limitation of working range in front of/behind the pole, 1st TRANSMIT. |

| MD24920 $MC_TRANSMIT_BASE_TOOL_1[ ] | Vector of the base tool for the 1st TRANSMIT transformation. |
|---|---|
| [0] = 0 | |
| [1] = 0 | |
| [2] = 0 | |

## TRANSMIT with real Y axis

| MD24100 $MC_TRAFO_TYPE_1 | |
|---|---|
| = 257 | Definition of transformation 1 in the channel: TRANSMIT main spindle. |

| MD24110 $MC_TRAFO_AXES_IN_1[ ] | Axis assignment for the 1st transformation in the channel. |
|---|---|
| [0] = 1 | Perpendicular to the rotary axis XC |
| [1] = 3 | Rotary axis C1 |
| [2] = 2 | Parallel to the rotary axis ZC |

| MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[ ] | Assignment of the geometry axes to channel axes for transformation 1. |
|---|---|
| [0] = 1 | 1st channel axis X |
| [1] = 3 | 2nd channel axis Y |
| [2] = 2 | 3rd channel axis Z |

| MD24900 $MC_TRANSMIT_ROT_AX_OFFSET_1 | |
|---|---|
| = 0 | Offset of the rotary axis for the 1st TRANSMIT transformation. |

| MD24905 $MC_TRANSMIT_ROT_AX_FRAME_1 | |
|---|---|
| = 2 | Axial offset of the rotary axis is taken into account during TRANSMIT 1. |

| MD24910 $MC_TRANSMIT_ROT_SIGN_IS_PLUS_1 | |
|---|---|
| = 0 | Sign of the rotary axis for the 1st TRANSMIT transformation. |

| MD24911 $MC_TRANSMIT_POLE_SIDE_FIX_1 | |
|---|---|
| = 1 | Limitation of working range in front of/behind the pole, 1st TRANSMIT. |

| MD24920 $MC_TRANSMIT_BASE_TOOL_1[ ] | Vector of the base tool for the 1st TRANSMIT transformation. |
|---|---|
| [0] = 0 | |
| [1] = 0 | |
| [2] = 0 | |

## Additional settings for rotary axes

If you want to integrate rotary axes, in addition to the following machine data you also require MD28082 with bit 6, see Chapter "Cylinder surface transformation (TRACYL) (Page 390)".

| MD24906 $MC_TRANSMIT_BASE_TOOL_COMP_1 | Compensation of the basic offset in the 1st transformation frame. |
|---|---|
| Bit 0 | 1st geometry axis |
| Bit 1 | 2nd geometry axis |
| Bit 2 | 3rd geometry axis |

| MD24956 $MC_TRANSMIT_BASE_TOOL_COMP_2 | Compensation of the basic offset in the 2nd transformation frame. |
|---|---|
| Bit 0 | 1st geometry axis |
| Bit 1 | 2nd geometry axis |
| Bit 2 | 3rd geometry axis |

For further information, see also Chapter "Clamping a blank (Page 144)".

## Face machining under ShopTurn

If you wish to use the face machining function under ShopTurn function, set the following additional channel-specific configuration machine data:

| MD52214 $MCS_FUNCTION_MASK_MILL | Function screens, ShopTurn. |
|---|---|
| Bit 3 | Enable "internal / rear" machining in ShopTurn screens that define the machining plane themselves. |
| Bit 4 | If you have implemented the "Clamp/release spindle" function using the machine manufacturer cycle CUST_TECHCYC.SPF, then using this machine data, you can activate the "Clamp/release spindle" parameter in the drilling and milling screens. |

| | |
|---|---|
| = 0 | The "Clamp/release spindle" parameter is not displayed in the drilling and milling screens. ShopTurn automatically clamps the spindle if it makes sense for the particular machining operation. |
| = 1 | The "Clamp/release spindle" parameter is displayed in the drilling and milling screens. The operator decides for which machining operation the spindle should be clamped. |

**Note**

The face end machining is automatically integrated in the cycles, with the exception of the straight line and circle.

You can select the functions for these two cycles in the "Program" operating area at "Straight line" and "Circle".

Additional information on end face machining is provided in:

### References

Function Manual Extended Functions; M1: Kinematic transformation,
Section: "TRANSMIT face end transformation"

## 18.5.10 Inclined Y axis (TRAANG)



**Software option**

You require the following software option in order to use the inclined Y axis function on your machine:

"Inclined axis"

### Function

If your turning machine has an inclined Y axis (i.e. this axis is not perpendicular to axes X and Z), you can still completely program machining operations in Cartesian coordinates. The control uses the inclined axis function (TRAANG) to transform the Cartesian coordinates to the motion of the inclined axis.

### Setting up

You still have to set up the inclined axis function (TRAANG) via machine data.

## Inclined axis under ShopTurn

Please also note the following:

**Note**

In the user interface, the inclined axis function is automatically integrated in the cycles after setting up. For machining with inclined axis, in the screen forms of the machining plane, you can select "Face Y" or "Peripheral surface Y" and enter the Cartesian coordinates.

You will find additional information on the inclined Y axis function in:

## References

- Function Manual Extended Functions; M1: Kinematic transformation, Section: "TRAANG oblique angle transformation"
- Programming Manual, Job Planning, Section "Transformations" > "Kinematic transformations" > "Inclined axis (TRAANG)"

## Example

For example, for a turning machine with X and Z axes and inclined Y axis, main spindle (C) and tool spindle (WZ), you must configure the following machine data:

| MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[ ] | Assignment of geometry axis to channel axis. |
|---|---|
| [0] = 1 | 1st real geometry axis X axis. |
| [1] = 0 | 2nd real geometry axis Y axis not available. |
| [2] = 2 | 3rd real geometry axis Z axis. |

| MD20110 $MC_RESET_MODE_MASK | |
|---|---|
| Bit 0 = 1<br>Bit 7 = 0 | TRAANG is retained after ramp-up. |

| MD20112 $MC_START_MODE_MASK | |
|---|---|
| Bit 7 = 1 | TRAANG is retained after "Cycle start". |

| MD20118 $MC_GEOAX_CHANGE_RESET | |
|---|---|
| = 1 | Allow automatic geometry axis change. |

| MD20140 $MC_TRAFO_RESET_VALUE | |
|---|---|
| = 5 | TRAANG always active after reset. |

| MD20144 $MC_TRAFO_MODE_MASK | |
|---|---|
| Bit 0 = 1 | TRAANG runs in the background (persistent) and is not shown on the user interface. |

| MD20070 $MC_AXCONF_MACHAX_USED[4] | |
|---|---|
| = 5 | Channel axis YC = 5th machine axis. |

| MD20080 $MC_AXCONF_CHANAX_NAME_TAB[ ] | Name of channel axis in the channel. |
|---|---|
| [0] = XC | 1st channel axis XC |
| [1] = ZC | 2nd channel axis ZC |
| [2] = C | 3rd channel axis C |
| [3] = WZ | 4th channel axis WZ |
| [4] = YC | 5th channel axis YC |

**Data set for inclined axis:**

| MD24430 $MC_TRAFO_TYPE_5 | |
|---|---|
| = 1024 | Transformation 5: TRAANG |

| MD24432 $MC_TRAFO_AXES_IN_5[ ] | Axis assignment for transformation 5. |
|---|---|
| [0] = 5 | 1st transformation axis = channel axis, YC |
| [1] = 1 | 2nd transformation axis = channel axis, XC |
| [2] = 2 | 3rd transformation axis = channel axis, ZC |

| MD24434 $MC_TRAFO_GEOAX_ASSIGN_TAB_5[ ] | Assignment of geometry axes to channel axes for transformation 5. |
|---|---|
| [0] = 1 | 1st axis = channel axis XC |
| [1] = 5 | 2nd axis = channel axis YC |
| [2] = 2 | 3rd axis = channel axis ZC |

| MD24436 $MC_TRAFO_INCLUDES_TOOL_5 | |
|---|---|
| = 0 | Tool handling with active transformation 5. |

| MD24700 $MC_TRAANG_ANGLE_1 | |
|---|---|
| = 55 | Angle between 1st and 2nd transformation axis. Data set for linking (TRACON) of end face machining on main spindle (TRANSMIT) and inclined axis (TRAANG). |

**Data set for linking (TRACON) of end face machining on main spindle (TRANSMIT) and inclined axis (TRAANG):**

| MD24440 $MC_TRAFO_TYPE_6 | |
|---|---|
| = 8192 | Type of transformation that is available as sixth in the channel. |

| MD24444 $MC_TRAFO_GEOAX_ASSIGN_TAB_6[ ] | Assignment of geometry axes to channel axes for transformation 6. |
|---|---|
| [0] = 1 | 1st axis = channel axis XC |
| [1] = 3 | 2nd axis = channel axis YC |
| [2] = 2 | 3rd axis = channel axis ZC |

| MD24995 $MC_TRACON_CHAIN_1[ ] | Transformation linking |
|---|---|
| [0] = 1 | Number of the TRANSMIT transformation (main spindle) for linking. |
| [1] = 5 | Number of the TRAANG transformation for linking. Data set for linking (TRACON) of cylinder surface transformation on main spindle (TRACYL) and inclined axis (TRAANG). |

**Data set for linking (TRACON) of cylinder surface transformation on main spindle (TRACYL) and inclined axis (TRAANG):**

| MD24450 $MC_TRAFO_TYPE_7 | |
|---|---|
| = 8192 | Type of transformation 7 in the TRACON channel. |

| MD24454 $MC_TRAFO_GEOAX_ASSIGN_TAB_7[ ] | Assignment of geometry axes to channel axes for transformation 7. |
|---|---|
| [0] = 1 | 1st axis = channel axis XC |
| [1] = 3 | 2nd axis = channel axis YC |
| [2] = 2 | 3rd axis = channel axis ZC |

| MD24996 $MC_TRACON_CHAIN_2[ ] | Transformation linking |
|---|---|
| [0] = 3 | Number of the TRACYL transformation (main spindle) for linking. |
| [1] = 5 | Number of the TRAANG transformation for linking. |

## 18.5.11 Spindle speed limitation for current program

Speed limits are programmed in the part program. If the part program is applied, the spindle speed limits are entered in the setting data and saved there after the end of program.

However, if the modified speed limits must not apply after the end of the program, you can change the property with the following commands in the GUD block of the machine manufacturer:

### Changing spindle speed limits for the current program only

```
REDEF $SA_SPIND_MAX_VELO_LIMS PRLOC    ; maximum spindle speed limit for V
                                       constant (G96)
REDEF $SA_SPIND_MAX_VELO_G26 PRLOC     ; maximum spindle speed limit for rpm
                                       (G26)
```

For further information about speed limitation, please refer to:

### References

- Programming Manual Fundamentals, Section "Spindle motion" > "Programmable spindle speed limitation (G25, G26)"
- Programming Manual Job Planning, Section "Flexible NC programming" > "Variables" > "Redefinition of system data, user data, and NC commands (REDEF)"

## 18.6 Grinding

### General

Grinding tools are identified by tool type 400, 410; dressing tools are identified by tool type 490. The "New tool" function offers these types for selection. The tool list shows the "Grinding disk" and "Dressing tool" tool types listed according to their tool type. To generate the grinding disk correction, a tool setter creates a dressing program. Depending on the complexity, the tool setter makes use of external or internal code generators. This program will be assigned later to a grinding disk.

### Oscillation cycles

These grinding cycles permit the machining of workpieces with grinding machines using two geometry axes.

The machines type does not exist because the cycles create only an oscillating movement that can be used as required on every machine.

This requires a SINUMERIK control as well as fast inputs/outputs for the program processing.

The cycles package provides the following cycles:

- CYCLE4071: Longitudinal grinding with infeed at the reversal point
- CYCLE4072: Longitudinal grinding with infeed at the reversal point and measurement control
- CYCLE4073: Longitudinal grinding with continuous infeed
- CYCLE4074: Longitudinal grinding with continuous infeed and measurement control
- CYCLE4075: Surface grinding with infeed at the reversal point
- CYCLE4077: Surface grinding, plunge grinding with measurement control
- CYCLE4078: Surface grinding with continuous infeed
- CYCLE4079: Surface grinding with intermittent infeed

## Oscillation cycles: Activate GUD variables for synchronized actions

| MD18661 $MN_MM_NUM_SYNACT_GUD_INT[n] | | Number of configurable integer-type GUD variables |
|---|---|---|
| > 0 | With the machine data, the individual GUD blocks can be extended by additional channel-specific parameter areas of the type integer.<br>[n] is equivalent to the access rights data block | |

## References

For further information about oscillating cycles and GUD variables for synchronized actions, please refer to:

- Programming Manual, Production Planning; Chapter "Programming cycles externally" > "Technological cycles"
- Function Manual Synchronized Actions, Chapter "Detailed description" > "User-defined variables in synchronized actions"

## Grinding function screen

| MD51840 $MNS_GRIND_FUNCTION_MASK | | Grinding cycles function screen |
|---|---|---|
| Bit 0 | Reference point for measuring the dresser/wheel is a work offset | |
| = 0 | Deselection of the work offset as reference point | |
| = 1 | Selection of the work offset as reference point | |

| MD52840 $MCS_FUNCTION_MASK | | Grinding cycles function screen |
|---|---|---|
| Bit 7 | Retraction when form-truing with G1 | |
| = 0 | Retraction when form-truing with rapid traverse (G0) | |
| = 1 | Retraction when form-truing with feedrate (G1) | |

## Activate and display fit-dependent corrections

Activate fit-dependent corrections

| MD18603 $MN_MM_NUM_GLOBAL_G_FRAMES | Number of global grinding frames (SRAM) |
|---|---|
| The value corresponds to the number of field elements for the pre-defined field $P_GFR[].<br>If the value of the data is > 0, then all settable frames are only global. MD28079 is then ignored. | |

| MD28079 $MC_MM_NUM_G_FRAMES | Number of grinding frames (SRAM) |
|---|---|
| Defines the number of predefined grinding frames. | |

Display fit-dependent corrections in the operating area "Parameters" → "Work offsets" window:

| MD52211 $MCS_FUNCTION_MASK_DISP_ZOA | | Function screen Display overview of work offsets |
|---|---|---|
| Bit 22 | Display fit-dependent corrections ($P_GFRAME) | |

## Details of disk size

| MD52842 $MCS_GRIND_DIAMETER_LENGTH | | Number of the tool length for the disk diameter |
|---|---|---|
| = 1 | Length 1 is disk diameter | |
| = 2 | Length 2 is disk diameter | |

| MD52843 $MCS_GRIND_WIDTH_LENGTH | | Number of the tool length for the disk width |
|---|---|---|
| = 1 | Length 1 is disk width | |
| = 2 | Length 2 is disk width | |

## Form-truing (CYCLE495)

| SD55880 $SCS_GRIND_CONT_RELEASE_ANGLE | | Angle of retraction |
|---|---|---|
| | This setting data is used to specify the angle of retraction from the contour during form-truing. | |

| SD55881 $SCS_GRIND_CONT_RELEASE_DIST | | Retraction distance |
|---|---|---|
| | This setting data is used to specify the distance retracted in both axes during form-truing. | |

| SD55884 $SCS_GRIND_CONT_BLANK_OFFSET | | Blank allowance |
|---|---|---|
| | This setting data is used to specify the distance to the grinding wheel as of which the switch-over is made from G0 to G1 during form-truing. | |

# 18.7 Swiveling

## 18.7.1 Technology cycles for swiveling

### Precondition

The commissioning of the machine kinematics is a mandatory requirement for correct swiveling functionality (CYCLE800). The vectors of the machine kinematics are stored in the tool parameters $TC_CARR1 to $TC_CARR65.

**Note**

The vectors of the machine kinematics can be determined with the "Measure kinematics" measuring function (CYCLE996).

To activate the swivel function, there must be at least one tool holder that can be oriented (swivel data set) in the NCK, and the workpiece, tool and rotary table reference system frames must be activated:

| MD18088 $MN_MM_NUM_TOOL_CARRIER | Maximum number of definable tool holders |
|---|---|
| > 0 | |

| MD28082 $MC_MM_SYSTEM_FRAME_MASK | System frames (SRAM) |
|---|---|
| = 3DH | |
| Bit 2 = 1 | TCARR and PAROT |
| Bit 3 = 1 | TOROT and TOFRAME |
| Bit 4 = 1 | Workpiece reference points |
| Bit 5 = 1 | System frame for cycles |

**Note**

Changing machine data MD18088 and MD28082 causes the buffered memory to be reorganized.

After changing the machine data, a series start-up file must be generated and downloaded, otherwise, it can be assumed that data will be lost.

To change machine data, you require the following authorization: Access level 1 (manufacturer).

You will find up-to-date information in the following references:

## References

- "siemensd.txt" file in the supplied software (standard cycles)
- Function Manual Basic Functions; W1: Tool offset
  Section: "Tool holder with orientation capability" > "Oblique machining with 3 + 2 axes"
- Programming Manual, measuring cycles: CYCLE996

## Activating the swivel function

The swivel function is enabled on the user interface via the following channel-machine data:

| MD52212 $MCS_FUNCTION_MASK_TECH | General function screen for all technologies |
|---|---|
| Bit 0 = 1 | Enable swivel |

## Configuring input screens

You can configure the input screen for swiveling using the following channel-specific cycle setting data. The setting data is effective for all of the declared swivel data sets.

| SD55221 $SCS_FUNKTION_MASK_SWIVEL_SET | Function screen, swiveling CYCLE800 |
|---|---|
| Bit 0 | Input field "No swiveling" |
| = 0 | Hide |
| = 1 | Display |
| Bit 1 | Select text when retracting "Z, ZY" or "Fixed position 1/2" |
| = 0 | Display text Z = "Z", display text Z, XY = "Z,XY" |
| = 1 | Display text Z = "Fixed point 1", Display text Z, XY = "Fixed point 2".<br>If you wish to modify the retraction version "Z" or "Z, XY" via the manufacturer cycle CUST_800.SPF, the neutral text "Fixed point 1" and "Fixed point 2" can be displayed. |
| Bit 2 | Select, permit "deselection" of the swivel data set |
| = 0 | If deselection is not permitted, the "Swivel data set" (TC) selection field is not displayed in the "Swivel" input screen. |
| = 1 | Deselection permitted<br>see also swivel data set parameter $TC_CARR37<br>HUNDRED MILLIONS position |
| Bit 3 | Displays the active swivel plane under swivel in JOG. The setting in the swivel function screen acts on all swivel data sets. |
| Bit 4 | Positioning in the basic position (pole position) of the kinematics |
| = 0 | Evaluation of the input values in pole position of the machine kinematics |
| = 1 | Compatibility |
| Bit 5 | Tool alignment absolute or via frame calculation |
| = 0 | Tool alignment via frame calculation (TCOFRY for G18) |
| = 1 | Tool alignment absolute (TCOABS)<br>For B axis kinematics of a turning machine, the tool spindle can be set in the basic position of the kinematics, either parallel to Z or to X. Therefore, the function "Align milling tool" is required for rotary milling machine bit 5 = 1. |
| Bit 6 | Do not list swivel mode "direct" under swivel in JOG |
| Bit 7 | Swivel plane: Direction selection in basic position of the kinematic |

| | |
|---|---|
| = 0 | With direction selection + or -, both calculated solutions for the rotary axes are approached |
| = 1 | With direction selection + or -, only one of the calculated solutions for the rotary axes is approached |
| Bit 8 | Traverse the zero offset for the rotary axes of the swivel data set as offset |
| = 0 | Only take the zero offset for the rotary axes of the swivel data set into account in the WCS (compatibility) |
| = 1 | Traverse the zero offset for the rotary axes of the swivel data set as offset<br>Bit 8 is set in conjunction with MD21186. |
| Bit 9 | Swivel data set TC permanently assigned to the tool |
| = 0 | Show number of the swivel data set TC |
| = 1 | Do not show number of the swivel data set TC<br>The swivel data set is assigned to the tool with parameter SGUD_TC_GNO if technology MD52200 = 3 is set for cylindrical grinding.<br>Example:<br>The machine manufacturer can write the current tool _TC_GNO = $P_TOOLNO into parameter _TC_GNO in the tool change program L6. As a result, the tool holder will be updated whenever a tool is changed. |
| Bit 10 | Swivel plane: Show input field "Positioning direction" |
| = 0 | Input field "Positioning direction" is not displayed |
| = 1 | Input field "Positioning direction" is displayed with swivel table and swivel head/table combinations<br>The selected positioning direction is stored in GUD _TC_POS_DIR and can be evaluated in CUST_800:<br>_TC_POS_DIR = 0: Automatic (shortest path)<br>_TC_POS_DIR = -1: Direction -<br>_TC_POS_DIR = +1: Direction +<br>The function must be implemented by the manufacturer. |
| Bit 11 | Positioning both solutions of AB kinematics |
| = 0 | Positions only one solution of some swivel angles (compatibility) |
| = 1 | Positions both solutions calculated by the NC<br>Bit 11 should be set for kinematics = 1 where, in the kinematics basic setting, neither of the two rotary axes rotates about the tool axis (AB kinematics in tool axis in the Z direction) |
| Bit 12 | Swivel plane, align tool: Reset behavior takes account of rotary axis positions |
| = 0 | Reset condition refers to the final position of the rotary axes of the swivel data set (compatibility)<br>Linear axis identifiers are written into parameters $TC_CARR21/22. On reset (TCOABS), the tool holder is initialized with the final positions of the rotary axes ($TC_CARR13/14). |
| = 1 | Reset condition refers to the current position of the rotary axes of the swivel data set<br>The rotary axis names of the swivel data set are written into parameters $TC_CARR21/22. On reset (TCOABS), the tool holder is initialized with the actual values of the rotary axes. |
| Bit 13 | Activates the extended evaluation of rotations for axis-by-axis swiveling in the pole position |
| = 0 | The rotations when swiveling in the pole position (compatibility) are not evaluated, the settings of bits 4 and 11 apply |
| = 1 | Extended evaluation of rotations when swiveling in the pole position |

## Additional settings

For the swivel function, set the following machine data as a minimum to the following - although this deviates from the default value:

| MD10602 $MN_FRAME_GEOAX_CHANGE_MODE | |
|---|---|
| = 1 | The actual total frame (zero offsets) is recalculated when switching over geometry axes (selecting/deselecting TRAORI). |

| MD10760 $MN_G53_TOOLCORR | Method of operation with G53, G153 and SUPA |
|---|---|
| Bit 0 = 1 | If the machine axis/axes are to be retracted before swiveling, see Chapter "Manufacturer cycle CUST_800.SPF (Page 435)" |

| MD11450 $MN_SEARCH_RUN_MODE | Settings, block search |
|---|---|
| Bit 1 = 1 | Activate PROG_EVENT.SPF after block search. This means that for a block search, the rotary axes of the active swivel data set are pre-positioned. |

## Declare swivel data sets

For the swivel function, swivel data sets must be created in the NCK and declared in the following machine data.

| MD18088 $MN_MM_NUM_TOOL_CARRIER | Maximum number of definable tool holders that can be oriented.<br>Number of swivel data sets |
|---|---|

If several channels are declared in the NCU, the number of swivel data sets is split up, taking MD28085 $MN_MM_LINK_TOA_UNIT into account.

**Example 1:**

Machine has two channels with different TO units.

Channel 1: MD28085 $MN_MM_LINK_TOA_UNIT= 1

Channel 2: MD28085 $MN_MM_LINK_TOA_UNIT= 2

Three swivel data sets (SDS) are required for each TO unit.

MD18088 = (number of TO units) x (number of SDS per TO unit) = 2 x 3 = 6

**Example 2:**

Machine has three channels with two different TO units.

Channel 1: MD28085 $MN_MM_LINK_TOA_UNIT= 1

Channel 2: MD28085 $MN_MM_LINK_TOA_UNIT= 2

Channel 3: MD28085 $MN_MM_LINK_TOA_UNIT= 1

One swivel data set (SDS) is required for each TO unit.

MD18088 = (number of TO units) x (number of SDS per TO unit) = 2 x 1 = 2

## Using angular tools

Angular tools are created and managed in the NC with tool type 130. The tool lengths are entered in the tool of tool type 130. If an angular tool is also used on a swiveled machining plane (CYCLE800), the basic tool orientation must be enabled with MD18114 $MM_ENABLE_TOOL_ORIENT = 2. This enables a direction vector to be transferred to the tool in the tool parameters $TC_DPV3[n] to $TC_DPV5[n]. The TOROT (G17), TOROTY (G18), and TOROTX (G19) commands must be programmed for the tool change so that the basic tool orientation is updated.

n = internal tool number

| MD18114 $MM_ENABLE_TOOL_ORIENT | Assign orientation to cutting edges.<br>Basic tool orientation |
|---|---|
| = 0 | No basic tool orientation active |
| = 2 | Basic tool orientation active<br>Used for angular tools and swiveling |

| MD20110 $MC_RESET_MODE_MASK | Initial control setting for RESET and end of part program |
|---|---|
| Bit 14 = 1 | Calculation of the basic and system frames, example: 4041H. |

| MD20112 $MC_START_MODE_MASK | Basic control setting for START and part program. |
|---|---|
| 400H | |

| MD20126 $MC_TOOL_CARRIER_RESET_VALUE | Active tool holder at RESET. |
|---|---|
| = 0 | No active tool holder at RESET. |
| = > 0 (n) | Tool holder with number n active at RESET.<br>MD20126 is written to in CYCLE800.<br>CYCLE800() corresponds to deselect tool holder (MD20126 = 0). |

| MD20150 $MC_GCODE_RESET_VALUES[ ] | Delete position, G group. |
|---|---|
| [41] = 1 | Delete position, G group 42 on TCOABS |
| [51] = 2 | Initial setting of G group 52 on PAROT |
| [52] = 1 | Delete position, G group 53 on TOROTOF |
| [52] = > 1 | Delete position, G group 53 on TOROT, TOROTY or TOROTX<br>Used for machine kinematics, types "T" and "M".<br>See parameter $TC_CARR34 |

**Note**

If, after a RESET from the NC, a frame must be calculated in the tool direction, then MD20150 $MC_GCODE_RESET_VALUES[52] can be set to a value > 1.

Applications:

- Machine kinematics with Hirth joint
- Angular tool with basic tool orientation

## Note on kinematics (swivel head / mixed kinematics) with Hirth joint:

Depending on the active plane (G17, G18, G19), the TOROT command (or TOROTX, TOROTY) is programmed in the NCU (G group 53) to calculate the compensating frame for the Hirth joint in CYCLE800. If the Hirth joint causes the programmed rotation to deviate from the possible positions of the rotary axes, a $P_TOOLFRAME compensating frame is created for swivel head and mixed kinematics (see HMI active ZO / details tool reference).

If the compensating frame must be retained after RESET or end of part program, enter the following value in the channel-specific machine data:

| MD20150 $MC_GCODE_RESET_VALUES[52] | Reset behavior of the G groups |
|---|---|
| = 2 | For G17 (TOROT) |
| = 3 | |
| = 4 | For G18 (TOROTY) |
| = 5 | For G19 (TOROTX) |

| MD20196 $MC_TOCARR_ROTAX_MODE | Rotary axis mode for oriented tool holders |
|---|---|
| Bit 0 = 1 | Swivel data set with a rotary axis<br>used for a rotary table with C axis |
| Bit 1 = 1 | Swivel data set with two rotary axes<br>used for standard swiveling |
| Bit 2 | |
| = 1 | In a singular position, for which there is an unambiguous solution due to the alignment of the coordinate axes that are vertical to the orientation, the value 1 is displayed in the system variable $P_TCSOL as the number of solutions. |
| = 0 | This version is required to ensure compatibility with older software versions. Two solutions are displayed; the angles about the orientation axis differ by 180 degrees. |

| MD20360 $MC_TOOL_PARAMETER_DEF_MASK | Tool parameter setting |
|---|---|
| Bit 10 = 1 | The orientation vector remains at T0 or D0 (no tool).<br>Used for machine kinematics, types "T" and "M".<br>See parameter $TC_CARR34 |
| Bit 19 = 1 | The changed orientation vector remains at T0 or D0 (no tool).<br>Bit 19 acts only with bit 10 = 1. |

| MD21186 $MC_TOCARR_ROT_OFFSET_FROM_FR | | Offset of the rotary axes for an oriented tool holder from the zero offset of the rotary axis. |
|---|---|---|
| = 0 | In CYCLE800, the WCS is recalculated when there is a value in the zero offset (ZO) of the rotary axes.<br>If values are entered in the zero offset of the rotary axes of the active swivel data set, the basic position of the swivel data set and therefore also the basic position of the WCS can change. | |
| = 1 | A value in the ZO of the rotary axes acts as offset of the oriented tool holder. The Work remains unchanged. | |

MD21186 may not be rewritten in a program with call CYCLE800.

The setting MD21186 = 0 is only suitable for a swivel data set in which one of the available rotary axes rotates exactly around the tool axis. In this way, a useful rotation of the WCS can be calculated in CYCLE800 when values are entered in the zero offset of the rotary axes. For example, for a swivel table with rotary axis A around X and rotary axis C around Z for a basic position of the kinematics in G17.

If several swivel data sets are declared per channel, and if machine functions need to be activated on changeover between swivel heads or tables, an M command can be issued in the PLC program on switchover to another swivel data set.

| MD22530 $MC_TOCARR_CHANGE_M_CODE | | M code for swivel data set change |
|---|---|---|
| = 0 | No swivel data set change | |
| < 0 | M code + number of the swivel data set for the swivel data set change | |

## Example

| Number of swivel data sets in channel 1 | = 2 |
|---|---|
| MD22530 $MC_TOCARR_CHANGE_M_CODE | = -800 |
| Programming swivel data set 1 (TCARR=1) | = M801 |
| Programming swivel data set 2 (TCARR=2) | = M802 |

With the output of the M commands, the PLC can limit or invert the spindle speed or clamp or release the rotary axes, for example.

## Setting the workpiece, tool and rotary table reference

Using the following machine data, you can set the workpiece, tool and rotary table reference system frames or you can influence the behavior of the system frames.

Application: System frames can be active after Reset or Power On, in order to retract a drill from a swiveled position without causing a collision, for example.

| MD24006 $MC_CHSFRAME_RESET_MASK | | Active system frames after RESET |
|---|---|---|
| Bit 4 | System frame workpiece reference | |
| = 0 | Not active | |
| = 1 | Remains active | |

| MD24007 $MC_CHSFRAME_RESET_CLEAR_MASK | Clear system frames after RESET |
|---|---|
| Bit 4 | System frame workpiece reference |
| = 0 | Do not delete |
| = 1 | Delete |

When used for measuring or swiveling in JOG, the workpiece reference must be active at RESET and not cleared (cascaded measuring).

| MD24006 $MC_CHSFRAME_RESET_MASK | Active system frames after RESET |
|---|---|
| Bit 4 = 1 | System frame for workpiece reference remains active after RESET |

| MD24007 $MC_CHSFRAME_RESET_CLEAR_MASK | Clear system frames after RESET |
|---|---|
| Bit 2 = 0 | Reserved, do not delete TCARR and PAROT |
| Bit 3 = 0 | Reserved, do not delete TOROT and TOFRAME |
| Bit 4 = 0 | Do not clear system frame workpiece reference after RESET |

| MD24008 $MC_CHSFRAME_POWERON_MASK | Reset system frames after power on. |
|---|---|
| Bit 2 | System frame rotary table reference (PAROT) |
| = 0 | Do not reset |
| = 1 | Reset |
| Bit 3 | System frame tool reference (TOROT,..) |
| = 0 | Do not reset |
| = 1 | Reset |
| Bit 4 | System frame workpiece reference |
| = 0 | Do not reset |
| = 1 | Reset |

| MD24080 $MC_USER_FRAME_POWERON_MASK | Settings for settable frames. |
|---|---|
| Bit 0 | |
| = 0 | Settable zero offset via power on not active. |
| = 1 | Last active settable zero offset remains active after power on if MD20152 $MC_GCODE_RESET_MODE[7] = 1. |

Application: Zero offset G5xx, including all rotations, should remain active after Power On.

| MD28082 $MC_MM_SYSTEM_FRAME_MASK | Setting-up system frames (SRAM) |
|---|---|
| Bit 2 = 1 | Rotary table reference ($P_PARTFRAME) |
| Bit 3 = 1 | Tool reference ($P_TOOLFRAME) |
| Bit 4 = 1 | Workpiece reference ($P_WPFRAME) |

| MD28083 $MC_MM_SYSTEM_DATAFRAME_MASK | Setting-up data management, system frames (SRAM) |
|---|---|
| Bit 2 = 1 | Rotary table reference ($P_PARTFRAME) |

| | |
|---|---|
| Bit 3 = 1 | Tool reference ($P_TOOLFRAME) |
| Bit 4 = 1 | Workpiece reference ($P_WPFRAME) |

| MD28085 $MC_MM_LINK_TOA_UNIT | Assigning a TO unit to a channel (SRAM) |
|---|---|
| If several channels are set-up, then the following can be set in the machine data:<br>Assignment of the TO units (tools and orientable tool holders) to the NC channel. One TO unit can be assigned to several channels.<br>- OR -<br>Each NC channel is assigned one TO unit. | |

Axial machine data for the modulo rotary axes of the swivel data set

| MD30455 $MA_MISC_FUNCTION_MASK | Axis functions |
|---|---|
| Bit 0 | Modulo rotary axis programming |
| = 0 | Programmed positions must be in the modulo range, otherwise an alarm is triggered (e.g. 0 to 359.999 degrees). |
| = 1 | When positions are programmed outside of the modulo range, no alarms are signaled. The position is modulo converted internally (e.g. -180 to +180 degrees). |
| Bit 2 | Positioning, rotary axis |
| = 0 | As programmed |
| = 1 | Along the shortest path<br>Application: With the setting bit 2 = 1, the rotary axis C travels along the shortest path for G90 with DC. For further information, refer to Chapter "Manufacturer cycle CUST_800.SPF (Page 435)". |

| MD32010 $MA_JOG_VELO_RAPID[AX]<br>AX = axis name | Rapid traverse in JOG<br>Rotary and machine axes that should travel for swiveling in JOG. |
|---|---|
| = 10000 | Rapid traverse in JOG mode for swiveling in JOG |

| SD42980 $SC_TOFRAME_MODE | Setting, frame definition for TOROT, PAROT |
|---|---|
| = 2000 | Swivel (default value) |

| SD42974 $SC_TOCARR_FINE_CORRECTION | Fine offset TCARR (swivel data set) |
|---|---|
| = 0 | No fine offset of the swivel data set vectors. |
| = 1 | Fine offset of the swivel data set vectors.<br>The parameters of the swivel data set from $TC_CARR41[n] n…number of the swivel data set and higher apply. |

### Swiveling in the JOG mode

Cycle alarms 62186 and 62187 can be hidden or displayed using the following cycle machine data:

| MD55410 $MC_MILL_SWIVEL_ALARM_MASK | Activate fault evaluation CYCLE800 |
|---|---|
| Bit 0 | Activates fault 61186 |
| = 0 | Hide fault 61186 "Active zero offset G%4 and base (base reference) contains rotations" (default setting). |
| = 1 | Display fault 61186 |
| Bit 1 | Activate fault 61187 |
| = 0 | Hide fault 61187 "Active base and base reference (G500) contain rotations" (default setting). |
| = 1 | Display fault 61187 |

## 18.7.2 CYCLE800 checklist for the identification of the machine kinematics

**Note**

Identification of the machine kinematics according to DIN 66217 or ISO 841-2001

This checklist does not claim to be complete.

- **Do the 3 linear axes of the machine that are active for the transformation form an orthogonal coordinate system?** Geometry axes XYZ
- **How many swivel kinematics does the machine have?**
  Combinations of two (or one) rotary axis and the three linear axes are always formed.
- **Which machine components are concerned?**
  Swivel head, swivel table or swivel head and rotary table.
- **What are the names of the rotary axes of the kinematics?**
  Manual rotary axes are permitted and do not have to be declared in the NC.
- **What is the 1st or 2nd d rotary axis of a swivel data set?**
  Rule: Rotary axis 2 is based on rotary axis 1. With mixed kinematics, rotary axis 1 is always the axis for the tool orientation.
- **Is the traversing direction of the linear axes and the rotary axes correct?** Right-hand rule
  Rule: If the linear axis or the rotary axis moves the workpiece, the direction of motion of the axis and also the sign of the rotary axis vector change.
- **What is the initial setting of the kinematics?**
  This defines the tool orientation and the plane G17, G18, G19.
- **Which rotary axis rotates around which axis of the coordinate system or the machine axis (axes)?**
  This defines the rotary axis vectors of the kinematics.
  Example 1:
  Head kinematics. Rotary axis 2 rotates around axis Y → rotary axis vector V2xyz = 0,1,0
  Example 2:
  Table kinematics. Rotary axis 1 rotates around axis X → rotary axis vector V1xyz = -1,0,0

## 18.7.3 Commissioning swivel data (machine kinematics)

### Defining a swivel data set

You must create a swivel data set for every swivel head, swivel table or each combination of both.

Swivel data sets can be declared in several channels.

A swivel data set comprises the parameters $TC_CARR1[n] to $TC_CARR65[n]
n = number of the swivel data set.

The parameters of the swivel data set ($TC_CARR1[n] to $TC_CARR65[n]) can be read-in and read-out in the startup operating area. Programming with appropriate value assignment is also possible in an NC program (manufacturer cycle). The parameters of the swivel data set are immediately effective after the program has started.

The following softkeys are assigned to the Swivel function in the "Program" → "Various" operating area:

| Milling technology | Turning technology | |
|---|---|---|
| "Swivel plane" | "Swivel plane" | |
| "Advance milling tool" | "Swivel tool" | → "Align turning tool" |
| | | → "Align milling tool" |
| | | → "Advance milling tool" |

The "Align turning tool" and "Align milling tool" softkeys are only displayed if the "B-axis kinematics" function has been activated via $TC_CARR37[n].

### Offset vectors I1 to I4

The vectors always contain three components which represent the reference to the machine axes (X, Y, Z). The positions in the kinematic chain are measured by the machine manufacturer; they are always relevant with respect to a swivel head / swivel table (swivel data set). Offset vectors I1 to I4 refer to the non-swiveled state of the rotary axes (machine kinematics basic setting).

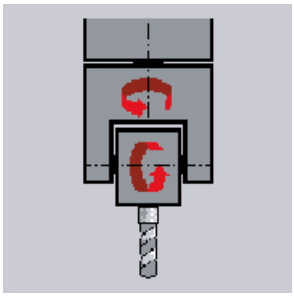
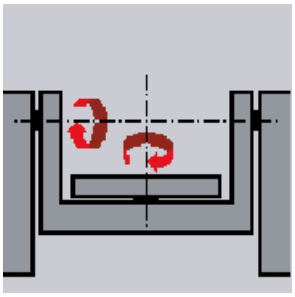
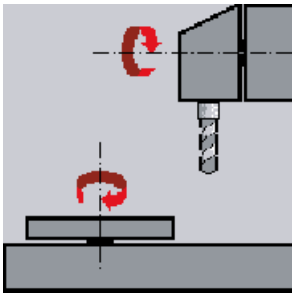
The machine kinematics used do not need to be fully implemented. However, be aware that the traversing range in the swivel planes may be restricted. If machine kinematics are to be implemented with just one rotary axis, this must always be declared as the 1st rotary axis.

| | | | |
|---|---|---|---|
| $TC_CARR1[n], | $TC_CARR2[n], | $TC_CARR3[n] | Offset vector I1xyz |
| $TC_CARR4[n], | $TC_CARR5[n], | $TC_CARR6[n] | Offset vector I2xyz |
| $TC_CARR15[n], | $TC_CARR16[n], | $TC_CARR17[n] | Offset vector I3xyz |
| $TC_CARR18[n], | $TC_CARR19[n], | $TC_CARR20[n] | Offset vector I4xyz |

### Rotary axis vectors V1 and V2

| | | | |
|---|---|---|---|
| $TC_CARR7[n], | $TC_CARR8[n], | $TC_CARR9[n] | Rotary axis vector V1xyz |
| $TC_CARR10[n], | $TC_CARR11[n], | $TC_CARR12[n] | Rotary axis vector V2xyz |

## Kinematic types $TC_CARR23[n]

| Swivel head (type T) | Swivel table (type P) | Swivel head + Swivel table (type M) |
|---|---|---|
|  |  |  |
| Offset vector I1 | Offset vector I2 | Offset vector I1 |
| Rotary axis vector V1 | Rotary axis vector V1 | Rotary axis vector V1 |
| Offset vector I2 | Offset vector I3 | Offset vector I2 |
| Rotary axis vector V2 | Rotary axis vector V2 | Offset vector I3 |
| Offset vector I3 | Offset vector I4 | Rotary axis vector V2 |
| | | Offset vector I4 |

## References

You can find additional information in:

- Function Manual Basic Functions; Tool Offset (W1)
- Function Manual Special Functions; Multiple Transformations (F2)

## Name of the swivel data set

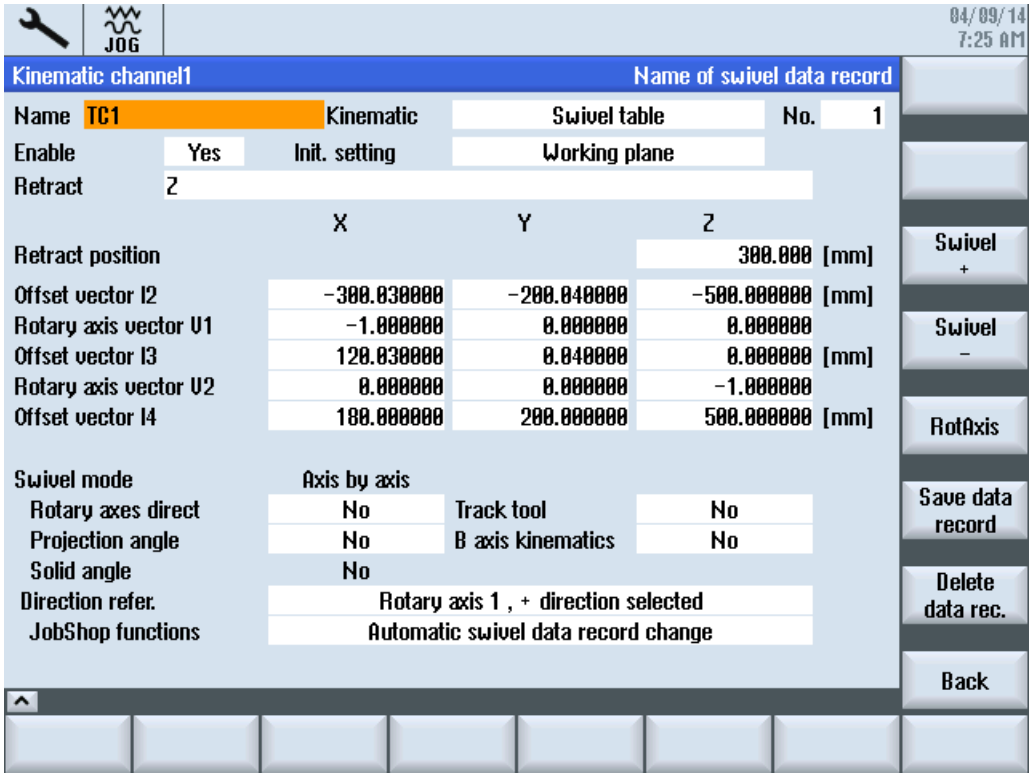



Figure 18-4 Dialog to input parameters for the swivel data set

If several swivel data sets are declared in each NC channel, then a name is assigned to each swivel data set: $TC_CARR34[n]

No name needs to be specified if the swivel-mounted tool holder is not exchangeable (i.e. one swivel data set per channel).

The name of the swivel data set may only contain characters that are permissible for NC programming: A...Z, 0...9 and _ !

If several NC channels have been set up, note the settings in MD28085, Chapter "Technology cycles for swiveling (Page 404)". If a TO unit is assigned to several NC channels, the channel display in the dialog corresponds to the TO unit.

## Offset vectors I1 to I4 $TC_CARR1[n] ... $TC_CARR20[n]

The vectors always contain three components which represent the reference to the machine axes (X, Y, Z). The positions in the kinematic chain are measured by the machine manufacturer; they are always relevant with respect to a swivel head/swivel table (swivel data record).

Offset vectors I1 to I4 refer to the non-swiveled state of the rotary axes (basic setting: machine kinematics). However, be aware that the traversing range in the swivel planes may be restricted. If machine kinematics are to be implemented with just one rotary axis, this must always be declared as the 1st rotary axis.

Manually adjustable rotary axes (manual mode) are possible with and without measuring systems and can be used with "simple machines".

Rotary axes that are positioned by an additional mechanical system, can be declared in the "Semiautomatic" mode.

### Example

In a boring mill, the swivel head is positioned by coupling the spindle. In this case, the rotary axes of the swivel head must be declared as "Semiautomatic". The appropriate machine-specific cycles are called in CUST_800.SPF.



**Machine manufacturer**
Please observe the information provided by the machine manufacturer.

### Swivel head

- I3 distance from the tool adapter to the pivot point/intersection of the 2nd rotary axis
- I2 distance from the pivot point/intersection of the 2nd rotary axis to the pivot point/ intersection of the 1st rotary axis
- I1 closure of the I1=-(I2+I3) vector chain, if the swivel head cannot be changed

### Swivel table

- I2 distance from the machine reference point to the pivot point/intersection of the 1st rotary axis
- I3 distance from the pivot point/intersection of the 1st rotary axis to the pivot point/ intersection of the 2nd rotary axis (or to the reference point of the tool adapter)
- I4 closure of the I4=-(I2+I3) vector chain, if the swivel table cannot be changed

### Swivel head/swivel table (mixed kinematics)

- I2 distance from the tool adapter to the pivot point/intersection of the 1st rotary axis
- I1 closure of the I1=-I2 vector chain, if the swivel head cannot be changed
- I3 distance from the machine reference point to the pivot point/intersection of the 2nd rotary axis (or to the reference point of the tool adapter)
- I4 closure of the I4=-I3 vector chain, if the swivel table cannot be changed

The offset vectors do not have to point to the pivot point of the rotary axes. The important thing is that they point to a point on the direction of rotation (intersection).

The sign of the offset vectors (I1 to I4) and the rotary axis vectors (V1, V2) result from the specifications of the axis directions according to **ISO 841-2001** or **DIN 66217** (right-hand rule).

In the case of machine kinematics that move the workpiece (rotary table), the axis direction is reversed.

**Note**

**Interrelation of TOOLCARRIER ⇔ 5-axis transformation (transformer type 24, 40, 56):**

For 5-axis transformation, transformer type 72 in MD24100: $MC_TRAFO_TYPE_1 can be used.

In the case of transformer type 72, the vectors of the TOOLCARRIER in MD24582: $MC_TRAFO5_TCARR_NO_1 are used.

| **Swivel head (swivel-mounted tool)** | | | |
|---|---|---|---|
| $TC_CARR23[n]="**T**" | | | MD24100: $MC_TRAFO_TYPE_1=**24** |
| **I1** | $TC_CARR1...3[n] | | MD24500: $MC_TRAFO5_PART_OFFSET_1 [0...2] |
| **I2** | $TC_CARR4...6[n] | | MD24560: $MC_TRAFO5_JOINT_OFFSET_1 [0...2] |
| **I3** | $TC_CARR15...17[n] | | MD24550: $MC_TRAFO5_BASE_TOOL_1 [0...2] |
| Closure of the **I1=-(I2+I3)** vector chain; for fixed-mounted machine kinematics | | | |

| **Swivel table (swivel-mounted workpiece)** | | | |
|---|---|---|---|
| $TC_CARR23[n]=**"P"** | | | MD24100: $MC_TRAFO_TYPE_1=**40** |
| **I2** | $TC_CARR4...6[n] | | MD24550: $MC_TRAFO5_BASE_TOOL_1 [0..2] |
| **I3** | $TC_CARR15...17[n] | | MD24558: $MC_TRAFO5_JOINT_OFFSET_1 [0...2] |
| **I4** | $TC_CARR18...20[n] | | MD24500: $MC_TRAFO5_PART_OFFSET_1 [0...2] |
| Closure of the **I4=-(I2+I3)** vector chain; for fixed-mounted machine kinematics | | | |

| **Swivel head + swivel table (swivel-mounted tool + workpiece)** | | | |
|---|---|---|---|
| $TC_CARR23[n]="**M**" | | | MD24100: $MC_TRAFO_TYPE_1=**56** |
| **I1** | $TC_CARR1...3[n] | | MD24560: $MC_TRAFO5_JOINT_OFFSET_1 [0...2] |
| **I2** | $TC_CARR4...6[n] | | MD24550: $MC_TRAFO5_BASE_TOOL_1 [0...2] |
| **I3** | $TC_CARR15...17[n] | | MD24558: $MC_TRAFO5_JOINT_OFF-SET_PART_1[0...2] |
| **I4** | $TC_CARR18...20[n] | | MD24500: $MC_TRAFO5_PART_OFFSET_1 [0...2] |
| Closure of the **I1=-I2 I4=-I3** vector chain; for fixed-mounted machine kinematics | | | |

| **Rotary axis vectors V1, V2** | | | |
|---|---|---|---|
| **V1** | $TC_CARR7..9[n] | | MD24570: $MC_TRAFO5_AXIS1_1[0..2] |
| **V2** | $TC_CARR10..12[n] | | MD24572: $MC_TRAFO5_AXIS2_1[0..2] |

## Fine offsets of offset vectors

| | | |
|---|---|---|
| $TC_CARR41[n] | to | $TC_CARR60[n] |

Assignment of base vectors to the fine offset vectors:

| | | |
|---|---|---|
| I1 $TC_CARR1..3[n] | to | $TC_CARR41..43[n] |
| I2 $TC_CARR4..6[n] | to | $TC_CARR44..46[n] |
| I3 $TC_CARR15..17[n] | to | $TC_CARR55..57[n] |
| I4 $TC_CARR18..20[n] | to | $TC_CARR58..60[n] |

The fine offsets are activated by the following setting data:

| SD42974 $SC_TOCARR_FINE_CORREC-TION = 1 | Fine offset TCARR on/off |
|---|---|
| = 0 | When activating a tool holder that can be oriented, the fine offset values are not taken into account. |
| = 1 | When activating a tool holder that can be oriented, the fine offset values are taken into account. |

The fine offsets act in addition to the corresponding base vectors when the Swivel function CYCLE800 or the NC function TCARR=n is called.

## Name of rotary axes

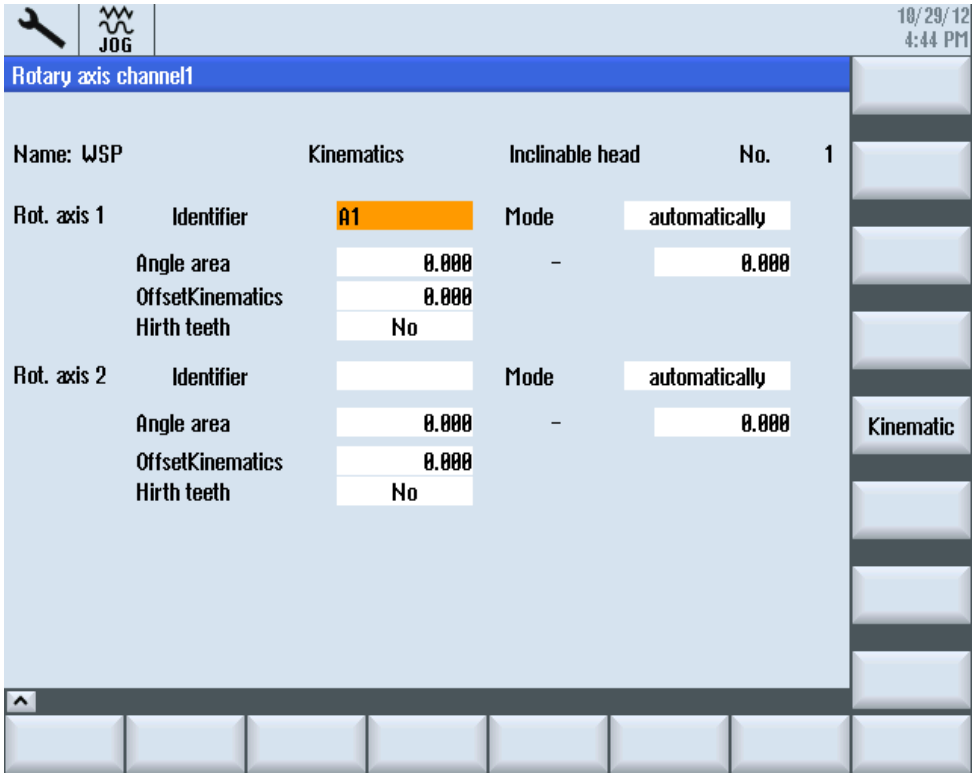



Figure 18-5 Dialog to input parameters for the rotary axis

The following identifiers should be preferably chosen for the names of the rotary axes:

- Rotary axis rotates around machine axis X → A
- Rotary axis rotates around machine axis X → B
- Rotary axis rotates around machine axis Z → C

For automatic rotary axes, the channel names of the corresponding NC rotary axes must be entered (see $TC_CARR37[n] TENS and HUNDREDS position: Automatic mode). For manual (manually adjustable) and semiautomatic rotary axes, you can use any axis identifier (up to six letters or digits).

Manual and semiautomatic rotary axes are not defined in the NC. With manual rotary axes, the value for the angular range is entered in the dialog box. Semiautomatic rotary axes are moved mechanically on the machine, e.g. through a coupled spindle. The adaptations for this can be made in the CUST_800 manufacturer cycle. The value for the angular range must not be specified, the rotary axis positioning is performed automatically.



**Machine manufacturer**
Please refer to the machine manufacturer's specifications.

| | |
|---|---|
| $TC_CARR35[n] | Name of rotary axis 1 |
| $TC_CARR36[n] | Name of rotary axis 2 |

## System variable $TC_CARR37[n]

In order that a value can be displayed in the input/selection boxes of the dialog for swiveling, the following display versions can be set:

### Meaning of the decimal positions

| **$TC_CARR37[n]** | **Display variants of the input screens for CYCLE800** | |
|---|---|---|
| **Decimal place** | **Meaning** | |
| ONES | Selects the swivel mode | |
| | 0 = | Axis-by-axis |
| | 1 = | Axis-by-axis + projection angle |
| | 2 = | Axis-by-axis + projection angle + solid angle |
| | 3 = | Axis-by-axis + direct |
| | 4 = | Axis-by-axis + projection angle + direct |
| | 5 = | Axis-by-axis + projection angle + solid angle + direct |
| TENS | Rotary axis 1 | |
| | 0 = | Automatic |
| | 1 = | Manual |
| | 2 = | Semiautomatic |
| HUNDREDS | Rotary axis 2 | |
| | 0 = | Automatic |
| | 1 = | Manual |
| | 2 = | Semiautomatic |

| **$TC_CARR37[n]** | **Display variants of the input screens for CYCLE800** | |
|---|---|---|
| **Decimal place** | **Meaning** | |
| THOUSANDS | Selection field direction: Direction selection of the rotary axes | |
| | 0 = | No display of the direction reference for kinematics that only have one solution. Direction selection (_DIR) Minus is generated in the cycle call CYCLE800. |
| | 3 = | Direction reference, rotary axis 1, direction selection Minus in the basic setting of the kinematics. |
| | 4 = | Direction reference, rotary axis 2, direction selection Minus in the basic setting of the kinematics. |
| | 5 = | No display of the direction reference for kinematics that only have one solution. Direction selection (_DIR) Plus is generated in the cycle call CYCLE800. |
| | 8 = | Direction reference, rotary axis 1, direction selection Plus in the basic setting of the kinematics. |
| | 9 = | Direction reference, rotary axis 2, direction selection Plus in the basic setting of the kinematics. |
| | | The values 1, 2, 6 and 7 are not permitted. |
| TEN THOUSANDS | Selection field, correction of the tool tip or B axis kinematics | |
| | 0 = | No display of the correction of the tool tip input field. |
| | 1 = | Correction of tool tip by means of TRAORI. |
| | 2 = | No correction of tool tip + B axis kinematics turning technology. |
| | 3 = | Correction of tool tip + B axis kinematics turning technology.<br>The Correction of tool tip function requires the "5-axis transformation (TRAORI)" option. |
| HUNDRED THOUSAND | Basic position of the kinematics [1)] | |
| | 0 = | Corresponding to the working plane G17, G18 or G19 (compatibility) |
| | 1 = | Tool axis in the Z direction |
| | 3 = | Tool axis in the X direction |
| ONE MILLION TEN MILLION | Selection field, retraction | |
| | 00 = | No retraction |
| | 01 = | Retraction Z |
| | 02 = | Retraction Z, XY |
| | 03 = | Retraction Z or Z, XY |
| | 04 = | Maximum retraction in tool direction |
| | ... | |
| | 08 = | Incremental retraction in tool direction |
| | ... | |
| | 15 = | Retraction Z or Z, XY or<br>in maximum tool direction or in incremental tool direction |
| | $TC_CARR38[n] | Retraction position X |
| | $TC_CARR39[n] | Retraction position Y |
| | $TC_CARR40[n] | Retraction position Z |

| $TC_CARR37[n] | Display variants of the input screens for CYCLE800 | |
|---|---|---|
| **Decimal place** | **Meaning** | |
| HUNDRED MILLION | Swivel data set enabled<br>Setting swivel data set change<br>Automatic or manual tool change is only necessary under ShopMill or ShopTurn (→ see also: CUST_800.spf, markers _M2 to _M13).<br>A swivel data set must be "enabled" in every case (value ≥ 4). | |
| | 0 = | Swivel data set not enabled |
| | 4 = | Swivel data set enabled<br>automatic swivel data set and tool change |
| | 5 = | Swivel data set enabled<br>automatic swivel data set change and manual tool change |
| | 6 = | Swivel data set enabled<br>manual swivel data set change and automatic tool change |
| | 7 = | Swivel data set enabled<br>manual swivel data set and tool change |

## Notes on the setting Basic position of the kinematics [1)]

For turning machines (1st technology is turning) with a B axis, you can set whether the tool is oriented toward "-Z" or toward "-X" in the basic position of the kinematics. This setting is taken into account for "Align tool" and for "Swivel plane".

### Align tool

- For "Align tool", the programming of beta and gamma is machine-independent.
- With the basic position setting "Working plane" or "-Z", beta = 0 also corresponds to this basic position (compatibility).
- With the basic position setting "-X", the B axis is re-oriented through 90° for beta = 0.

### Swivel plane (new swivel plane)

- With the basic position setting "Working plane" or "-Z" and active working plane G17, there is no new tool orientation for swivel plane to basic position (compatibility).
- With the basic position setting "-X" and active working plane G17, the tool orientation is rotated 90° around Y for swivel plane to basic position.

Therefore, with the basic position setting "-X", milling is possible in a swiveled plane in different working planes (G17 or G19).

The following settings and programming is not permitted and results in error messages:

- Swivel plane, basic position -Z and G18 or G19 active
- Swivel plane, basic position -X and G18 active

## Retracting the geometry axes before swiveling

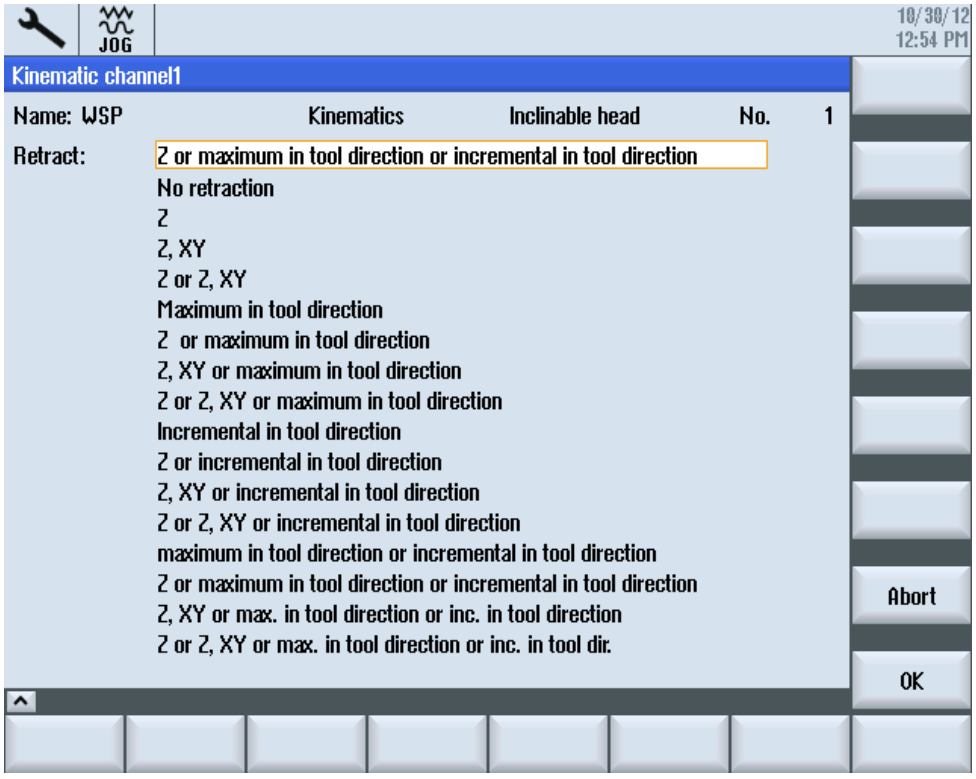



Figure 18-6 Dialog to select the type of retraction

Select the retraction variant in the input field: This is defined in the ONE MILLION and TEN MILLION position of the system variable $TC_CARR37[n]:

- Retract axis Z
- Retract axes Z, XY
- Retract in the tool direction, maximum or incremental

| NOTICE |
| --- |
| **Collision avoidance**<br>Make sure that the tool and the workpiece cannot collide during swivelling when moving the tool axes. |

Retracting axis Z or retracting axes Z, XY is realized as an absolute machine position at the values of parameters $TC_CARR38[n] to $TC_CARR40[n].

| | |
| --- | --- |
| $TC_CARR38[n] | Retraction position X |
| $TC_CARR39[n] | Retraction position Y |
| $TC_CARR40[n] | Retraction position Z |

The type of retraction is modified in the Manufacturer cycle CUST_800.SPF (Page 435).

**Note**

Retraction in the tool direction is useful for swivel data sets in which the tool is re-oriented (swivel head or mixed kinematics). This applies particularly for turning machines with a B axis. During retraction in "maximum tool direction", traversing is performed until one of the relevant linear axes reaches the software limit switch.

## Backing up swivel data set

With the "Save data set" softkey in the dialog for the swivel data set, all of the entered parameters ($TC_CARR1[n] to $TC_CARR65[n]) can be saved in a file in the part program memory. The file name corresponds to the name of the swivel data set. Observe the permitted characters when entering the swivel data set name.

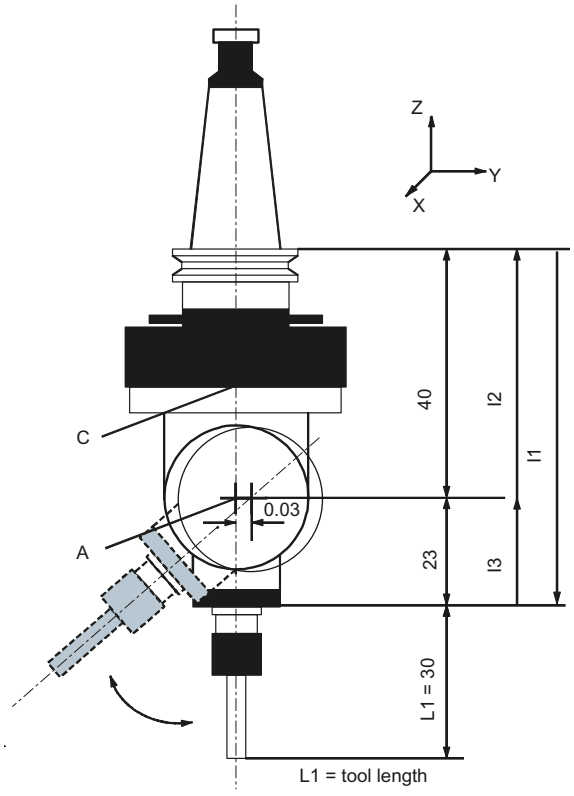
**Note**

All of the vectors and parameters immediately take effect after being entered and do not have to be additionally activated or saved.

## 18.7.4 Examples of machine kinematics for the commissioning of the Swivel function

### Example 1: Swivel head 1 "HEAD_1"

The vectors in the drawing refer to the initial setting of the kinematics. The drawing is not true to scale. The swivel head is exchangeable and manually adjustable.




C Rotary axis 1 (manual) rotates about Z.

A Rotary axis 2 (manual) rotates about X.

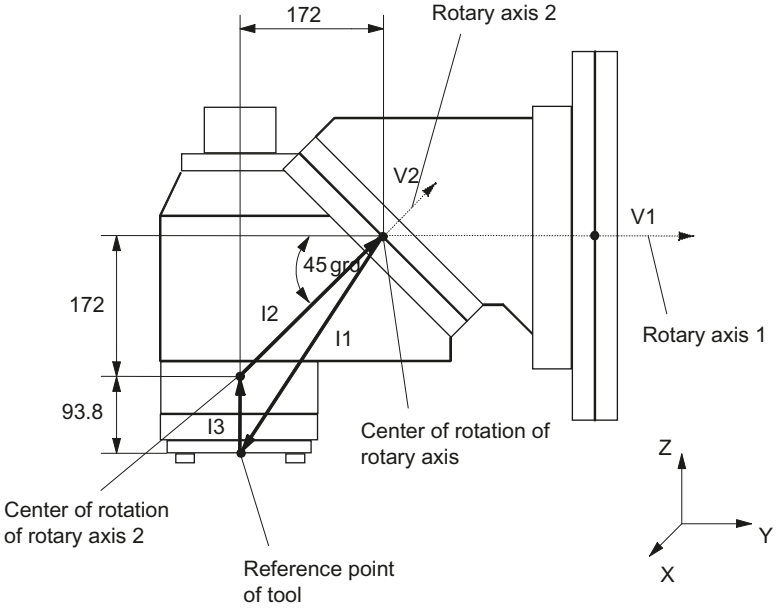
Figure 18-7 Exchangeable swivel head with steep taper to hold the spindle

| **Kinematics** | **Swivel head** | | **HEAD_1** |
|---|---|---|---|
| Retract | Z | | |
| | X | Y | Z |
| | | | 200.000 |
| Offset vector I1 | 0.000 | 0.030 | -63.000 |
| Rotary axis vector V1 | 0.000 | 0.000 | 1.000 |
| Offset vector I2 | 0.000 | 0.000 | 40.000 |
| Rotary axis vector V2 | 1.000 | 0.000 | 0.000 |
| Offset vector I3 | 0.000 | -0.030 | 23.000 |
| **Display version** | | | |
| Swivel mode | Axis-by-axis | | |
| Direction reference | Rotary axis 2 | | |

| Kinematics | Swivel head | | HEAD_1 |
|---|---|---|---|
| Correct tool | No | | |
| **Rotary axes** | | | |
| Rotary axis 1 | C | Mode | Manual |
| Angular range | 0.000 | | 360.000 |
| Rotary axis 2 | A | Mode | Manual |
| Angular range | -15.000 | | 100.000 |

## Example 2: Swivel head 2 "HEAD_2"

The vectors in the drawing refer to the initial setting of the kinematics. If the swivel head is fixed-mounted, the vector chain is closed. The reference point of the center of rotation of rotary axes 1 and 2 can be offset on the line of rotation and does not have to coincide with the mechanical center of rotation.




| | |
|---|---|
| Rotary axis vector V1 | Rotary axis 1 rotates about Y. |
| Rotary axis vector V2 | Rotary axis 2 rotates about Y and about Z. |
| Offset vector I1 | Closure of vector chain with fixed-mounted swivel head, I1 = - (I2 + I3). |
| Offset vector I2 | The distance between the center of rotation of rotary axis 1 and center of rotation of rotary axis 2 |
| Offset vector I3 | The distance between the reference point of the tool and the center of rotation of rotary axis 2. |

Figure 18-8 Cardanic swivel head with Hirth joint, manually adjustable

| Kinematics | Swivel head | | HEAD_2 |
|---|---|---|---|
| Retract | Z | Tool direction | Max.+inc. |
| | X | Y | Z |
| | | | 200.000 |
| Offset vector I1 | 0.000 | -172.000 | -265.800 |

| **Kinematics** | **Swivel head** | | **HEAD_2** |
|---|---|---|---|
| Rotary axis vector V1 | 0.000 | 1.000 | 0.000 |
| Offset vector I2 | 0.000 | 172.000 | 172.000 |
| Rotary axis vector V2 | 0.000 | 1.000[1)] | 1.000[1)] |
| Offset vector I3 | 0.000 | 0.000 | 93.800 |
| **Display version** | | | |
| Swivel mode | Axis-by-axis | | |
| Direction reference | Rotary axis 2 | | |
| **Rotary axes** | | | |
| Rotary axis 1 | B | Mode | Manual |
| Angular range | 0.000 | | 360.000 |
| Kinematics offset | 0.000 | | |
| Hirth joint | Yes | Angular grid | 1.000 |
| Rotary axis 2 | C | Mode | Manual |
| Angular range | 0.000 | | 180.000 |
| Kinematics offset | 0.000 | | |

1) Calculation of rotary axis vector:

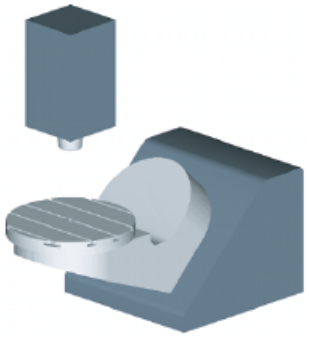
V2: 45 degree angle

V2Y = sin(-45) = -0.7071

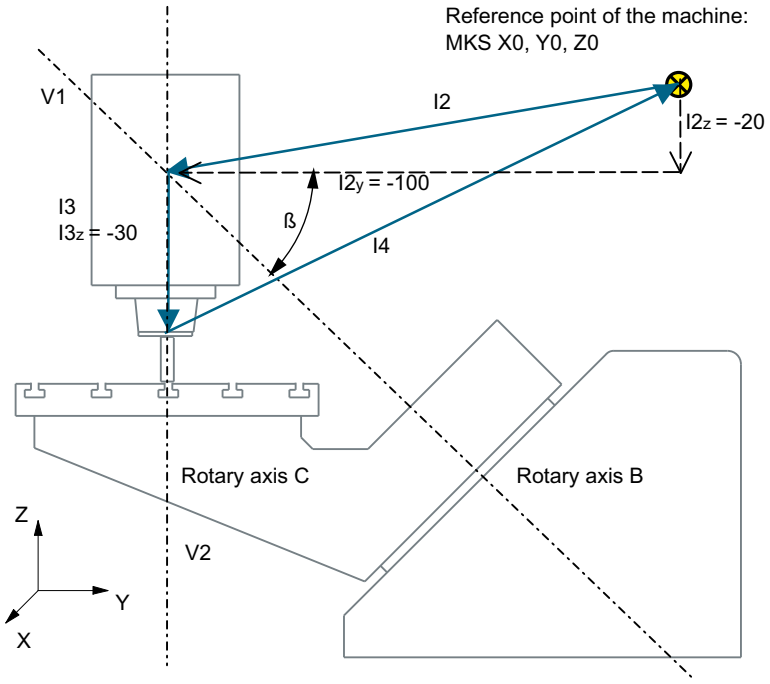
V2Z = cos(-45) = 0.7071

V2Y and V1Z can be normalized to 1.

## Example 3: Cardanic table "TABLE_45"



The vectors in the drawing refer to the initial setting of the kinematics. The spindle (tool adapter) is positioned on a gage block above the top edge of the table/center of the table (rotary axis C). A test mandrel in the spindle is used to determine the center of rotation of rotary axis C.




| | |
|---|---|
| Rotary axis vector V1 | Rotary axis B rotates around Y **and** around Z. |
| Rotary axis vector V2 | Rotary axis C rotates about Z. |
| Offset vector I2 | The distance from the reference point of the machine to the center of rotation/ intersection of rotary axis 1. |
| Offset vector I3 | The distance from the center of rotation/intersection of rotary axis 1 to the center of rotation/intersection of rotary axis 2. |
| Offset vector I4 | Closure of vector chain, I4 = - (I2 + I3). |

Figure 18-9 Side view of the machine

| **Kinematics** | **Swivel table** | | **TABLE_45** |
|---|---|---|---|
| | X | Y | Z |
| Offset vector I2 | 0.000 | -100.000 | -20.000 |
| Rotary axis vector V1 | 0.000 | -1.000[1)] | 1.000[1)] |
| Offset vector I3 | 0.000 | 0.000 | -30.000 |
| Rotary axis vector V2 | 0.000 | 0.000 | -1.000 |
| Offset vector I4 | 0.000 | 100.000 | 50.000 |
| **Display version** | | | |
| Swivel mode | Axis-by-axis | | |
| Direction reference | Rotary axis 2 | | |
| Correct tool | No | | |
| **Rotary axes** | | | |
| Rotary axis 1 | B | Mode | Auto |
| Angular range | 0.000 | | 180.000 |

| Kinematics | Swivel table | | TABLE_45 |
|---|---|---|---|
| Rotary axis 2 | C | Mode | Auto |
| Angular range | 0.000 | | 360.000 |

1) Calculation of rotary axis vector:

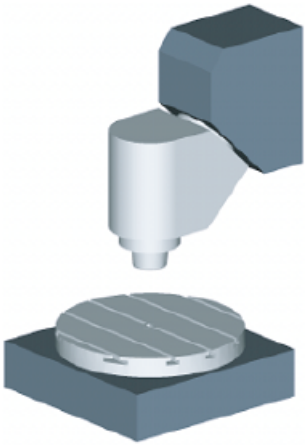
V1: β = -45 degrees

V1Y= sin(-45)= -0.7071

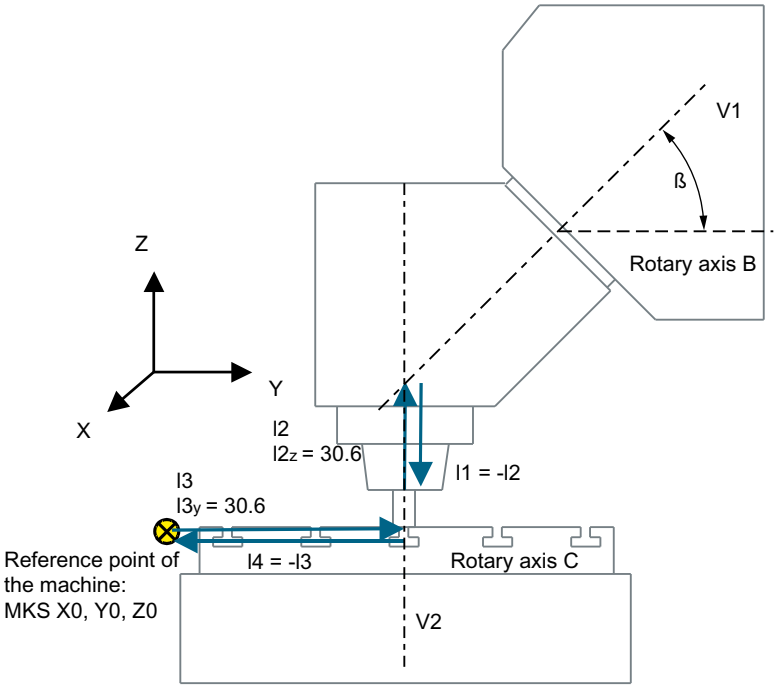
V1Z= cos(-45)= 0.7071

V1Y and V1Z can be normalized to -1 and 1.

## Example 4: Swivel head/rotary table "MIXED_45"



The vectors in the drawing refer to the initial setting of the kinematics. The spindle (tool adapter) is positioned on a gage block above the top edge of the table/center of the table (rotary axis C). A measuring rod in the spindle is used to determine the turning center of rotary axis C.




| | |
|---|---|
| Rotary axis vector V1 | Rotary axis B rotates around Y **and** around Z. |
| Rotary axis vector V2 | Rotary axis C rotates about Z. |
| Offset vector I2 | The distance from the reference point of the tool adapter to the center of rotation/intersection of rotary axis 1 |
| Offset vector I1 | Closure of vector chain, I1 = - I2. |
| Offset vector I3 | The distance from the reference point of the machine to the center of rotation/ intersection of rotary axis 2. |
| Offset vector I4 | Closure of vector chain, I4 = - I3. |

Figure 18-10 Side view of the machine

| **Kinematics** | **Mixed kinematics** | | **MIXED_45** |
|---|---|---|---|
| | X | Y | Z |
| Offset vector I1 | 0.000 | 0.000 | -30.600 |
| Rotary axis vector V1 | 0.000 | 1.000[1)] | 1.000[1)] |
| Offset vector I2 | 0.000 | 0.000 | 30.600 |
| Offset vector I3 | 300.000 | 150.000 | 0.000 |
| Rotary axis vector V2 | 0.000 | 0.000 | -1.000 |
| Offset vector I4 | -300.000 | -150.000 | 0.000 |
| **Display version** | | | |
| Swivel mode | Axis-by-axis | | |
| Direction | Rotary axis 1 | | |
| Correct tool | Yes | | |
| **Rotary axes** | | | |
| Rotary axis 1 | B | Mode | Auto |
| Angular range | 0.000 | | 180.000 |

| Kinematics | Mixed kinematics | | MIXED_45 |
|---|---|---|---|
| Rotary axis 2 | C | Mode | Auto |
| Angular range | 0.000 | | 360.000 |

1) Calculation of rotary axis vector:

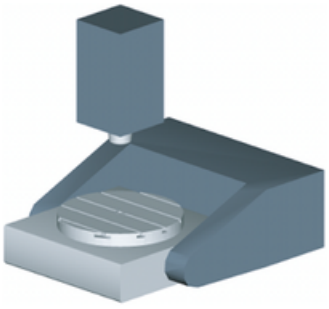
V1: β = 45 degrees

V1Y= sin(-45)= -0.7071

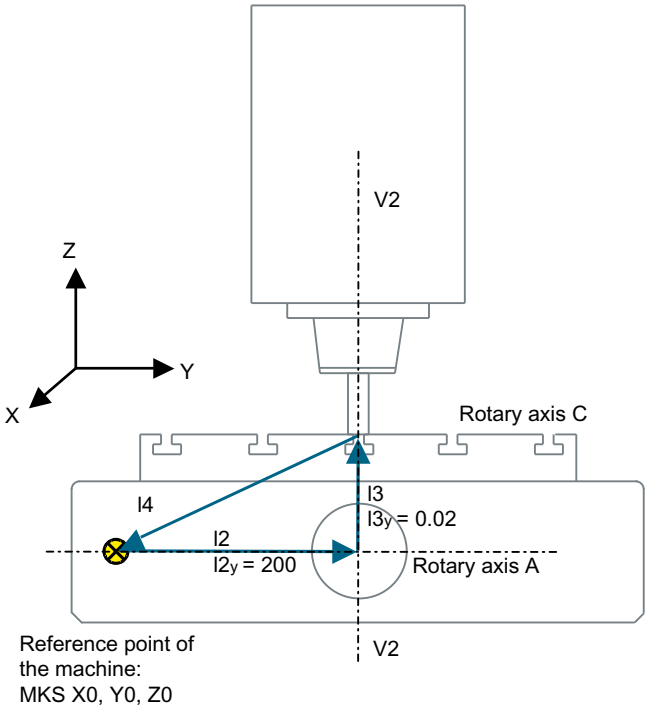
V1Z= cos(-45)= 0.7071

V1Y and V1Z can be normalized to 1.

## Example 5: Swivel table "TABLE_5"

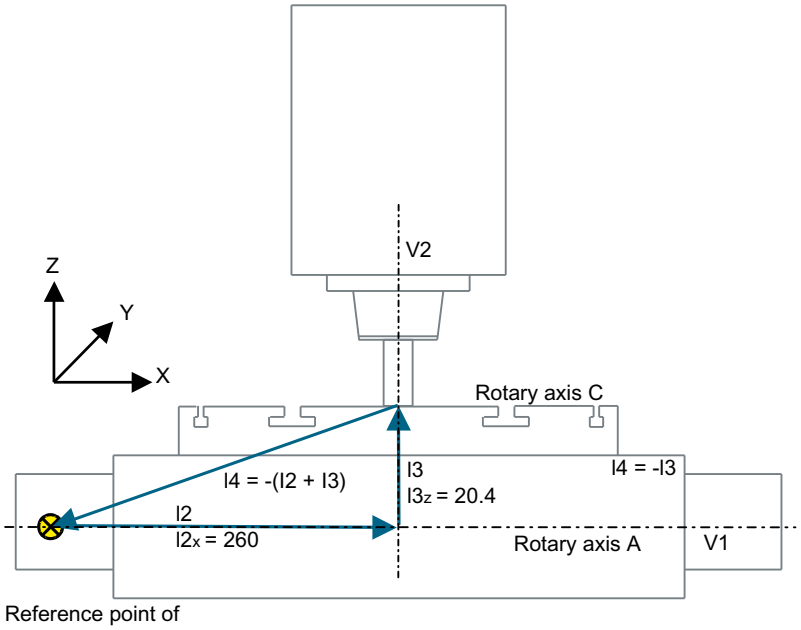


The vectors in the drawing refer to the initial setting of the kinematics. The spindle (tool adapter) is positioned on a gage block above the top edge of the table/center of the table (rotary axis C). A measuring rod in the spindle is used to determine the turning center of rotary axis C.




| | |
|---|---|
| Rotary axis vector V2 | Rotary axis C rotates about Z. |
| Offset vector I2 | The distance from the reference point of the machine to the center of rotation/ intersection of rotary axis 1. |
| Offset vector I3 | The distance from the center of rotation of rotary axis 1 to the center of rotation/intersection of rotary axis 2. |
| Offset vector I4 | Closure of vector chain, I4 = - (I2 + I3). |

Figure 18-11 Side view of the machine from the X direction




| | |
|---|---|
| Rotary axis vector V1 | The rotary axis A rotates about X. |
| Rotary axis vector V2 | Rotary axis C rotates about Z. |
| Offset vector I2 | The distance from the reference point of the machine to the center of rotation/ intersection of rotary axis 1. |
| Offset vector I3 | The distance from the center of rotation of rotary axis 1 to the center of rotation/intersection of rotary axis 2. |
| Offset vector I4 | Closure of vector chain, I4 = - (I2 + I3). |

Figure 18-12 Front view of the machine from the Y direction

| **Kinematics** | **Swivel table** | | **TABLE_5** |
|---|---|---|---|
| | X | Y | Z |
| Offset vector I2 | 260.000 | 200.000 | 0.000 |
| Rotary axis vector V1 | -1.000 | 0.000 | 0.000 |
| Offset vector I3 | 0.000 | 0.020 | 20.400 |
| Rotary axis vector V2 | 0.000 | 0.000 | -1.000 |
| Offset vector I4 | -260.000 | -200.020 | -20.400 |
| **Display version** | | | |
| Swivel mode | Axis-by-axis | | |
| Direction | Rotary axis 1 | | |
| Correct tool | No | | |
| **Rotary axes** | | | |
| Rotary axis 1 | A | Mode | Auto |
| Angular range | -90.000 | | 90.000 |
| Rotary axis 2 | C | Mode | Auto |
| Angular range | 0.000 | | 360.000 |

### Example 6: Setting up a turning machine with B axis

For turning machines with an additional rotary axis B, the tool can be aligned or swiveled on the X/Z plane. If your machine has a counterspindle, you can, for example, machine with a tool alternating between the main spindle and counterspindle.

You require the following swivel data sets to use the "Align tool" and "Swivel plane" functions:

- Swivel data set 1 for aligning and swiveling tools on the main spindle and counterspindle (turning, face Y/C, surface Y/C).
- Swivel data set 2 for machining inclined surfaces when milling on the main spindle (face B).
- Swivel data set 3 for machining inclined surfaces when milling on the counterspindle (face B).

When commissioning the "B-axis kinematics turning technology" in parameter $TC_CARR37[n], the TEN THOUSANDS position must be set to 2 or 3 in swivel data set 1. This is the reason why in the program editor "Swivel tool → align turning or milling tool" is listed for swivel data set 1.

See also Section "Commissioning swivel data (machine kinematics) (Page 414)"

### Example 7: Configuring a milling machine with B axis for turning.

You can configure a milling machine with a B axis (rotation around Y) so that turning is possible.

The following example refers to a milling machine with a rotary axis around Y (B), a rotary axis around Z (C) and a main spindle (SP) in the tool direction Z (G17).

You require the following swivel data sets to use the "Align tool" and "Swivel plane" functions:

- Swivel data set 1 for "Swiveling in the plane" when milling.
  Mixed kinematics: Rotary axis 1: B-axis, rotary axis 2: C axis
- Swivel data set 2 for the "Swivel tool - align tool" when turning
  Swivel head: Rotary axis 1: B-axis, rotary axis 2: Spindle SP

When milling, the SP spindle is the master spindle. Rotary axis C is declared as the master spindle in rotational operation, e.g. SETMS(2).

When commissioning the "B-axis kinematics turning technology" in parameter $TC_CARR37[n], the TEN THOUSANDS position must be set to 2 or 3 in swivel data set 2. This is the reason why in the program editor "Swivel tool → align turning or milling tool" is listed for swivel data set 2.

See also Section "Commissioning swivel data (machine kinematics) (Page 414)"

### References

Function Manual Basic Functions; W1: Tool offset, parameter CUTMOD:

- The CUTMOD NC function is activated when data set 2 is called in the NC program (CYCLE800 align turning tool). This means that the cutting edge position or the tool reference point is updated corresponding to the tool orientation:
- When calling a swivel data set for rotational operation in the NC program (CYCLE800 align tool), the CUTMOD NC function is activated. The cutting edge position, tool angle, cut direction and tool reference point are thus calculated in accordance with the current tool orientation following the "Align tool" function.

## 18.7.5 Manufacturer cycle CUST_800.SPF

### Function

During swiveling, all axis positions are approached using the CUST_800.SPF cycle. The call is exclusively made from the swivel cycle CYCLE800 or from the cycles E_TCARR (ShopMill) or F_TCARR (ShopTurn).

In cycle CUST_800.SPF, the function markers (_M2: to _M59) are prepared and documented. Also refer to the following "Structogram CYCLE800".

There are extensions in the CUST_800.SPF cycle for the "Turning on milling machine" function, see Chapter "Adaptations for CUST_800 (Page 454)".

**Note**

If you modify the CUST_800, ensure that the G commands of the G groups used are restored at the end of the CUST_800 (versions S_G_1 to S_G_5!).

### Parameter

CUST_800 (INT _MODE, INT _TC1, REAL _A1, REAL _A2, INT _TC2, REAL _T_POS)

| | |
|---|---|
| _MODE | A jump is made to markers _M2 to _M59 |
| _TC1 | Number of the swivel head/table |
| _A1 | Angle of rotary axis 1 |
| _A2 | Angle of rotary axis 2 |
| _TC2 | Feed evaluation in percent (%) for swiveling in JOG mode and number of the new swivel head/table when replaced in ShopMill |
| _T_POS | Incremental position during retraction in the incremental tool direction (see marker _M44, _M45) |

## Retract prior to swiveling

If the CUST_800.SPF cycle is not modified, the Z axis (marker _M41) or the Z axis followed by the X, Y axes (marker _M42) are first traversed in the Machine to the positions when retracting prior to swiveling. The freely available position values are specified in the system variables $TC_CARR38[n] to $TC_CARR40[n]. When retracting, the active tool cutting edge is deselected (D0) and is reselected after retraction.

If retraction in the tool direction has been declared, the tool axis is retracted to the software end position (maximum in tool direction) or by an incremental distance away from the tool in the tool direction. The tool lengths are taken into account accordingly.

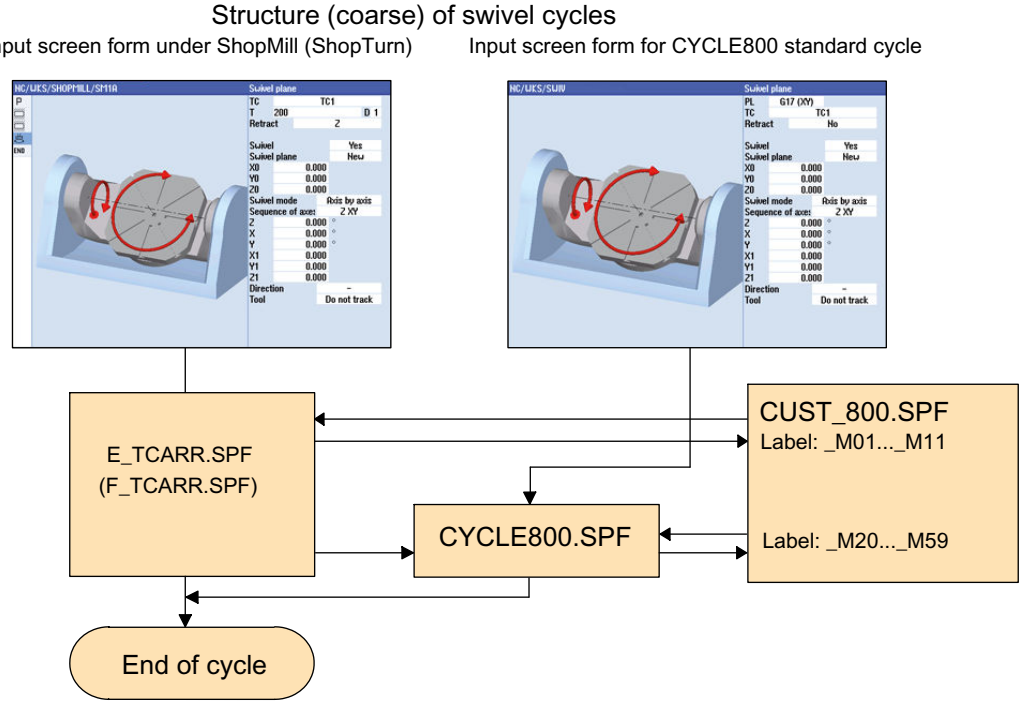



Figure 18-13 Structure (coarse) of swivel cycles

## CYCLE800 structured chart

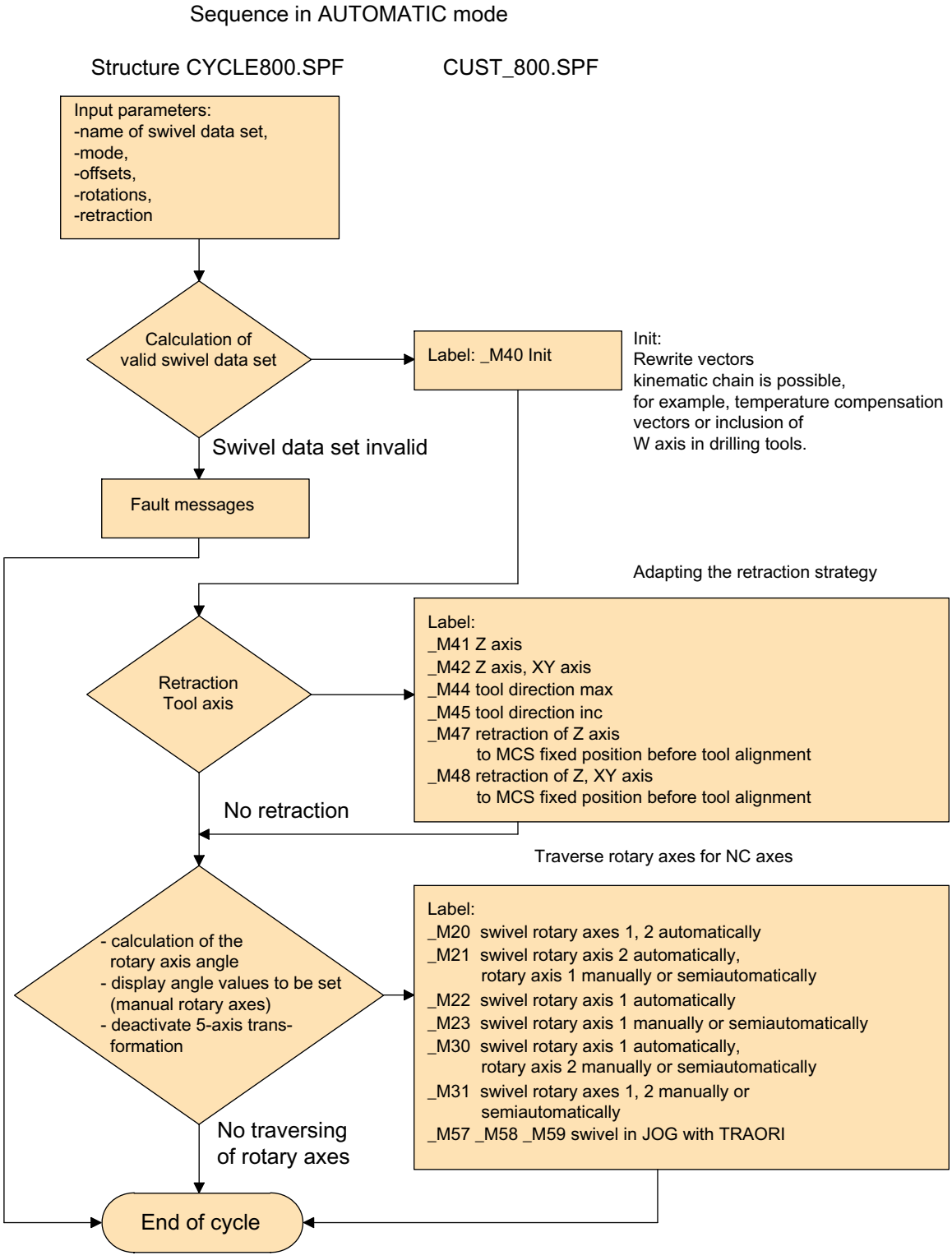



Figure 18-14 Structure: CYCLE800.SPF / CUST_800.SPF

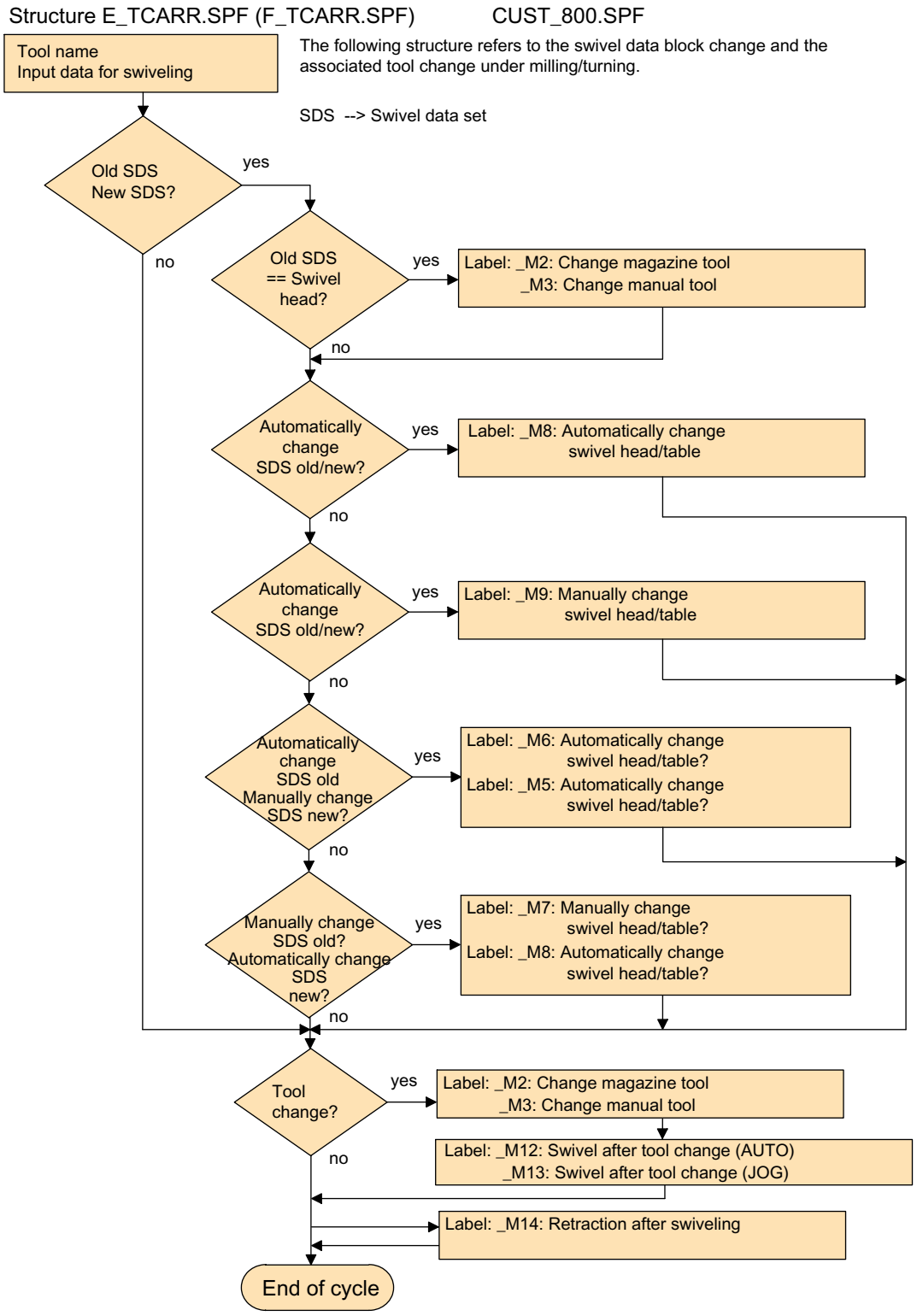



Figure 18-15 Structure: E_TCARR.SPF (F_TCARR.SPF) / CUST_800.SPF

### Note on markers _M2 to _M13

When the swivel data set or tool is changed, the linear axes are recovered using the last retraction mode (modal).

If this behavior is not desired in milling/turning, the corresponding calls must be commented out with a semicolon (;). The E_SWIV_H or F_SWIV_H cycle is called in milling/turning (see markers _M2 to _M9) in the CUST_800.SPF manufacturer cycle.

Parameter E_SWIV_H (Par1, Par2, Par3)

- Par1: Number of swivel data set (_TC1)
- Par2: Angle 1st rotary axis
- Par3: Angle 2nd rotary axis

### Modification examples

If the rotary axes (swivel head/table) are not to be positioned during swivel data change / tool change, the call of the E_SWIV_H cycle can be commented out at the relevant markers. If the rotary axes are to move to a certain position, an angle value can be transferred to parameters Par 2, Par 3.

### Note on labels _M14, _M15

Depending on the values of the retraction plane and the programmed swivel plane, it is possible that the linear axes now also travel the swiveled retraction plane while running up from the current position to the software limit switches after a block search. To avoid this problem, label _M14 in the CUST_800.SPF is called after swiveling. The E_SP_RP(30) cycle preset there runs up to the milling retraction plane, whereby travel may be along the software limit switches. An appropriate retraction after block search can be set at label _M15.

### Note on the label _M16

If for a swivel data set change or when de-selecting the swivel data set, you wish to traverse the rotary axes to zero, the syntax of the _M16 label can be modified accordingly, e.g. activate block N8006 E_SWIV_H(_TC1,0,0).

### Note on labels _M20 to _M31

Markers _M20 to _M31 are distinguished by machine kinematics with two rotary axes or one rotary axis. A distinction is also made between automatic rotary axes (known to the NCU) and manual (semi-automatic) rotary axes. There is only ever one valid label for swiveling with the active swivel data set.

### Note on label _M35

Run through _M35 for block search and a swivel data set with manual rotary axes.

### Note on label _M40

This label is called before the calculation and activation of the tool holder function as user initialization. This makes it possible for special kinematics, to calculate or rewrite the current vectors of the active swivel data set.

Boring mill application example: For parallel axes, the vector in the Z direction can be updated with the actual value of the W axis.

A STOPRE is integrated for this label.
Manufacturers can test as to whether this STOPRE for label M40 in CUST_800.spf complies with their functional requirements. In this case, one must weigh up between velocity (without STOPRE) and functionality for interrupt conditions (RESET) for a swivel plane. This also depends on the type of machine kinematics.

### Note on labels _M41 to _M45 and _M47, _M48

At marks _M41 to _M45, the retraction strategy can be modified before the swiveling of the rotary axes and at marks _M47 and _M48 before the alignment of the tool.

The retraction to the MCS positions Z or Z, XY is made considering the tool length components without tool cutter deselection.

### Note on labels _M41 and _M42

When retracting in Z or Z, XY with mixed swivel head kinematics, the NC takes account of the kinematics reference point. If the head is tilted, positions are approached that might deviate from the values of the retraction position $TC_CARR38 to $TC_CARR40.

If you do not want this, you can deactivate and then reactivate the current tool holder at the labels _M41/_M42 and in the calculations of the tool components in cycle CUST_800.SPF.

Example:

```
IF (_MODE==41)OR(_MODE==42)
TCARR=0 ;deactivate tool holder
 IF $P_GG[6]==1
 ...
 ENDIF
TCARR=_TC1 ;reactivate tool holder
ENDIF
_M41: ;axis Z MCS fixed point
...
TCARR=0 ;deactivate tool holder
 IF($MN_G53_TOOLCORR .....
 ...
 ENDIF
TCARR=_TC1 ;reactivate tool holder
```

### Note on label _M46

Retraction before swiveling after a block search can be set at label _M46. Variable _E_VER is 1 if it is a milling technology program.

### Note on labels _M57 to _M59

Labels _M57 to _M59 are used for swiveling in JOG mode and active 5-axis transformation (TRAORI).

### Note on "Correct tool"

"Correct tool" requires that a 5-axis transformation is set up which is equivalent to the corresponding swivel data set. The programming section for "Correct tool" is integrated in the markers _M20, _M21, _M22 and _M30. The first 5-axis transformation is called with TRAORI(1).

### Note on tool change + swivel

In general, the swivel (CYCLE800) and tool change functions for a machine are independent of each other. Thus, the swiveled work plane can be retained in a technological sequence with multiple tools, e.g. centering, drilling, tapping.

If the rotary axes of the active swivel data set are involved in the mechanical sequence of the tool change or have to be retracted, this must be taken into account in the tool change program. After the tool change, the rotary axis positions are approached as prior to the tool change. If linear axes (geometry axes) are also involved in the tool change, the rotations in the NC (swivel frame) must not be deleted. Rather, the linear axes can be positioned as machine axes using the G153 or SUPA commands.

### Note on swiveling without active tool compensation

If swiveling the rotary axes without active tool cutting edge (D0) is not possible, then you can adapt this in cycle CUST_800.SPF:

```
_M40:
IF ((NOT $P_TOOL) AND _TC1)
 LOOP
 MSG ("No tool cutting edge active")
 M0
 STOPRE
 ENDLOOP
ENDIF
GOTOF_MEND
```

## 18.7.6 Indirectly programming the name of the swivel data set

### Function

The **_TC_CHANGE** GUD variable defined in the channel is a string with a length of 32. Using this variable, the user can determine before the CYCLE800 call which swivel data set is called in the CYCLE800, irrespective of what is programmed in the call itself.

The _TC_CHANGE GUD variable is re-initialized with each RESET. At the end of the CYCLE800, the variable is also deleted. This means that the variable must be re-described before each new CYCLE800 call.

The swivel data set is changed in the CYCLE800 following the CUST_800 jump to the _M40 marker. This means that the variable can also be described in the CUST_800.

### Programming example

```
N10 _TC_CHANGE="HEAD_2"
N20 CYCLE800(0,"HEAD_1",100000,57,0,0,0,0,0,0,0,0,-1,100,1)
N30 M0 ;* HEAD_2 is active

N40 _TC_CHANGE="HEAD_3"
N50 CYCLE800(0,"HEAD_1",100000,57,0,0,0,0,0,0,0,0,-1,100,1)
N60 M0 ;* HEAD_3 is active

N70 CYCLE800(0,"HEAD_1",100000,57,0,0,0,0,0,0,0,0,-1,100,1)
N80 M0 ;* HEAD_1 is active

N90 _TC_CHANGE="TABLE_2"
N100 CYCLE800(0,"TABLE_1",100000,57,0,0,0,0,0,0,0,0,-1,100,1)
N110 M0 ;* TABLE_2 is active
N120 M2
```

**Note**

The _TC_CHANGE variable must be re-described before each new CYCLE800 call.

### Programmed restrictions in the CYCLE800

The swivel data set named in _TC_CHANGE is checked for the following properties:

- The swivel data set to be activated must be enabled.
- The following parameters must match the swivel data set selected on the screen:
  - $TC_CARR23 (swivel data set type)
  - $TC_CARR37 (swivel mode, units digit)
  - $TC_CARR37 (selection of follow-up, ten-thousands digit)
  - The mode set (Bit 6, 7) on the screen (axis-by-axis/solid angle/projection angle/direct) to TC_CARR37

## 18.7.7 CYCLE996 measuring workpiece kinematics

### Function

With the Measure kinematics function, it is possible to calculate the geometric vectors used to define the 5-axis transformation (TRAORI and TCARR) by measuring the position of the ball in space.

The measurement is essentially carried out by means of workpiece probes, which scan three positions of a measuring ball on each rotary axis. The ball positions can be defined in

accordance with user specifications so that they correspond to the geometric ratios on the machine. The only way of setting the ball positions is to reposition the rotary axis that is to be measured in each case.

Aside from the basic mechanics of the machine, no specific knowledge is required to use CYCLE996. No dimension drawings or machine location diagrams are necessary to carry out measuring.

## Application range

The Measure kinematics function (CYCLE996) can be used to determine the data that is relevant to transformations in the case of kinematic transformations that involve rotary axes (TRAORI, TCARR).

Options:

- Redetermination of swivel data sets
  - Commissioning the machine
  - Use of swivel-mounted workholders as TCARR
- Checking swivel data sets
  - Service following collisions
  - Checking the kinematics during the machining process

Kinematics with manual axes (manually adjustable rotary tables, swivel-mounted workholders) can be measured in the same way as kinematics with NC-controlled rotary axes.

When CYCLE996 is started, a swivel data set with basic data (for kinematics type) must be parameterized. The measurement itself must be carried out without an active kinematic transformation.

If, in the setting data SD55740 $MNS_MEA_FUNCTION_MASK bit 7 is set, then depending on the setting, the kinematics can be measured based on the calculated or on the rotary axis vectors saved in the swivel data set.

## Requirements

The following requirements must be met in order to use CYCLE996 (Measure kinematics):

- SIEMENS measuring cycles package is installed
- Workpiece probe is calibrated
- Calibration ball is mounted
- Oriented tool holder is set up (MD18088 MM_NUM_TOOL_CARRIER > 0)
- The basic geometry of the machine (X, Y, Z) is rectangular and referenced
  The right angle refers to the workpiece spindle and should be preferably checked using a test mandrel.
- Defined position of the rotary axes involved in the transformation
- Defined traverse directions in compliance with the standard of all axes involved in the transformation according to ISO 841-2001 and/or DIN 66217 (righthand rule)
- Defined size of the protocol file (MD11420 $MN_LEN_PROTOCOL_FILE ≥ 20)

The precise procedure when measuring and programming, including examples, is described in the following manual:

### References

Programming Manual, Measuring Cycles: CYCLE996

## 18.8 Turning on milling machines

You can set up the turning technology as an extended technology on milling machines. The operating software has all the setting options of a milling machine and a turning machine. For example, the rotary table can be set at a high speed for the turning of the workpiece. The turning tool is then clamped in the milling spindle and can be used in various positions.



**Machine manufacturer**
Please refer to the machine manufacturer's specifications.

### 18.8.1 Setting up machine data and setting data

#### Setting up parameters

**Machine data**

Set the following machine data to set up the technology:

| MD10602 $MN_FRAME_GEOAX_CHANGE_MODE | Frames when switching over geometry axes |
|---|---|
| = 1 | The current complete frame is calculated again when the geometry axes are switched over, and the translations, scalings and mirrorings of the new geometry axes are effective. The rotations of the old geometry axes are retained. |

| MD18114 $MN_MM_ENABLE_TOOL_ORIENT | Assign orientation to cutting edges |
|---|---|
| = 0 | The tool orientation function is not active. |
| | - OR - |
| = 3 | Each tool cutting edge D = m of tool T = n is assigned to the following system parameters:<br>$TC_DPV[n, m] and $TC_DPV3 - $TC_DPV5<br>$TC_DPVN3[n, m], $TC_DPVN4[n, m] and $TC_DPVN5[n, m] |

**Note**

If you use angular tools (type 130) with variable tool orientation, you require the setting MD18114 = 3.

| MD51226 $MNS_FUNCTION_MASK_SIM | | Simulation function screen |
|---|---|---|
| Bit 5 = 1 | Turning tool in tool spindle rotatable (for simulation) | |

| MD51781 $MNS_MEA_T_PROBE_THICKNESS | | Tool probe thickness for the automatic approach of turning tools to the probe on milling/turning machines |
|---|---|---|
| = 0 | Default value | |

| MD20100 $MC_DIAMETER_AX_DEF | | Geometry axis with transverse axis function |
|---|---|---|
| =X | The machine data defines a geometry axis as transverse axis. | |

| MD20110 $MC_RESET_MODE_MASK | | Definition of control initial setting after reset / end of part program |
|---|---|---|
| Bit 0 = 1 | Default value for system | |

| MD20112 $MC_START_MODE_MASK | | Definition of the basic control setting after part program start |
|---|---|---|
| Bit 7 = 0 | The current setting for the active transformation is retained. | |

| MD20124 $MC_TOOL_MANAGEMENT_TOOL-HOLDER | | Tool holder number |
|---|---|---|
| = 1 | Variable | |

| MD20125 $MC_CUTMOD_ERR | | Alarm display for CUTMOD errors |
|---|---|---|
| = CFFFH | Display alarms for CUTMOD errors | |

| MD20150 $MC_GCODE_RESET_VALUES[21] | | Initial setting of the G groups |
|---|---|---|
| = 2 | Initial setting of G group 22 on CUT2DF | |

| MD20150 $MC_GCODE_RESET_VALUES[55] | | Initial setting of the G groups |
|---|---|---|
| = 3 | Initial setting of G group 56 TOWWCS | |

| MD20152 $MC_GCODE_RESET_MODE[21] | | Reset behavior of the G groups |
|---|---|---|
| = 0 | Reset behavior of G group 22 | |

| MD20152 $MC_GCODE_RESET_MODE[55] | | Reset behavior of the G groups |
|---|---|---|
| = 3 | Reset behavior of G group 56 TOWWCS | |

| MD20360 $MC_TOOL_PARAMETER_DEF_MASK | Definition of the tool parameters |
|---|---|
| Bit 17 = 0 | With CUTMOD for turning and grinding tools, the cutting edge for the calculation of the compensation values is not configured in the machining plane. |
| Bit 18 = 1 | With CUTMOD for turning and grinding tools, always use the active plane for turning and grinding tools. |
| Bit 22 = 1 | Consider tool wear with active kinematic transformation in real time. |

| MD21203 $MC_LIFTFAST_MODE_MASK | Settings for fast lifting |
|---|---|
| Bit 0 = 1 | This bit activates the monitoring of the direction for LIFTFAST for thread turning. |

| MD24030 $MC_FRAME_ACS_SET | SZS coordinate system setting |
|---|---|
| = 1 | SZS results from the WCS transformed with the $P_CYCFRAME |

| MD35040 $MA_SPIND_ACTIVE_AFTER_RESET[ ] | Reset behavior of the spindle |
|---|---|
| [Tool spindle] = 2 | Do not stop tool spindle during jog → spindle is listed in the T,S,M screen |
| [Turning spindle] = 2 | Do not stop turning spindle during jog → spindle is listed in the T,S,M screen |

| MD52001 $MCS_DISP_COORDINATE_SYSTEM_2 | Definition of the displayed directions in the user interface for turning operations (variable) |
|---|---|
| = 0 | |

| MD52200 $MCS_TECHNOLOGY | Technology |
|---|---|
| = 2 | Milling |

| MD52201 $MCS_TECHNOLOGY_EXTENSION | Extended technology |
|---|---|
| = 1 | Turning |

| MD52206 $MCS_AXIS_USAGE[ ] | Meaning of the axes in the channel |
|---|---|
| [Tool spindle] = 1 | Tool spindle |
| [Turning spindle] = 3 | Turning spindle |

| MD52207 $MCS_AXIS_USAGE_ATTRIB[turning spindle] | Attributes of the axes |
|---|---|
| Bit 8 = 1 | Offer rotary axis for blank clamping |

| MD52212 $MCS_FUNCTION_MASK_TECH | General function screen for all technologies |
|---|---|
| Bit 10 = 1 | Align the turning tool via kinematics transformation |

| MD52214 $MCS_FUNCTION_MASK_MILL | Milling function screen |
|---|---|
| Bit 1 = 1 | Offer fixed table for blank clamping |

**Note**

Only set bit 1 in MD52214 when a fixed table is used in addition to a rotary table.

| MD52780 $MCS_J_MEA_T_PROBE_APPR_MODE | Approach mode for measuring a tool in JOG |
|---|---|
| = 0 | Technology-dependent approaching of the tool probe (default value)<br>Turning technology: Manual approach<br>Milling technology: Automatic approach |
| = 1 | Technology-independent approaching of the tool probe<br>Manual approaching of the tool probe |
| = 2 | Technology-independent approaching of the tool probe<br>Automatic approaching of the tool probe |

| MD52242 $MCS_TURN_TOOL_FIXING | Tool adapter for turning tools |
|---|---|
| = 0 | Turning tools in tool spindle |
| = 1 | Turning tools fixed on the tool headstock |
| = 2 | Turning tools in the tool spindle or fixed on the tool headstock |

**Note**

Depending on the set mounting alignment in MD52242, different parameters are available in the "Align Turning Tool" dialog.

### Setting data

The following setting data is recommended. Set the setting data to set up the technology:

| NOTICE |
|---|
| **Risk of collision**<br>Value changes of SD42940, SD42942 and SD42950 during the program execution can cause collisions. Only set the setting data during the commissioning. |

| SD42935 $SC_WEAR_TRANSFORM | Transformations for tool components |
|---|---|
| = 1 | Orientation for milling tools: Standard<br>G group 56: Observe or set TOWWCS |

| SD42940 $SC_TOOL_LENGTH_CONST | Change of tool length components at change of plane |
|---|---|
| = 17 | L1 = Z, L2 = Y, L3 = X |

| SD42942 $SC_TOOL_LENGTH_CONST_T | Change of tool length components for turning tools at change of plane |
|---|---|
| = 19 | L1 = Z, L2 = Y, L3 = X |

**Note**

SD42942 is only evaluated after setting SD42950.

| SD42950 $SC_TOOL_LENGTH_TYPE | Assignment of the tool length offset independent of tool type |
|---|---|
| = 3 | The assignment of the tool length components is performed separately for the turning and grinding tools (tool types 400 to 599) on the one hand and all other tools (milling tools) on the other hand. |

| SD42954 $SC_TOOL_ORI_CONST_M | Change of tool orientation components for milling tools at change of plane |
|---|---|
| = 0 | Orientation for milling tools: Standard |

| SD42956 $SC_TOOL_ORI_CONST_T | Change of tool orientation components for turning tools at change of plane |
|---|---|
| = 18 | Orientation for turning tools: G18 |

| SD42998 $SC_CUTMOD_PLANE_TOL | Difference between tool tip plane and machining plane |
|---|---|
| = 5 | Maximum permissible difference for gamma of 0° to 180° |

| SD42999 $SC_ORISOLH_INCLINE_TOL | Difference between tool tip plane and machining plane for ORISOLH |
|---|---|
| = 1 | Maximum permissible inclination of the tool tip compared to the G18 plane for "Align tool directly" |

In the following setting data, index [k] stands for the number of the current data field (probe number -1) of the probe.

| SD54633 $SNS_MEA_TP_TYPE[k] | Probe type, cube/disk |
|---|---|
| = 0 | Compatibility (measuring cycles: cube, turning surface shows probe, milling surface shows disk) |
| = 3 | Cube |

| SD54648 $SNS_MEA_TPW_TYPE[k] | Probe version |
|---|---|
| = 0 | Compatibility (measuring cycles: cube, turning surface shows probe, milling surface shows disk) |
| = 3 | Cube |

### Additional settings

- Set the current zero point in the center point of the turning tool, e.g. via a settable zero offset.
- Change the tool before the "Align turning tool" cycle call.
- Program the tool in the part program for the simulation.

  **Note**

  No tool is active after the start of the simulation.

- After the "Align turning tool" cycle call, traverse the tool to Y = 0.
- Limit the maximum speed of the tool spindle, e.g. with the tool change cycle, in order to avoid too high a speed of the turning tool.
- The turning tools in the tool management are automatically set up by the system. You can make your own modifications in file sltmmillturnlistconfig.xml, for example, for the OEM tool list.

## 18.8.2 Setting up a kinematics transformation

In order to be able to use the turning on milling machines technology, at least one kinematics transformation must be set up for turning.

### Requirement

A kinematic chain is already available for the machine.

Information on kinematic elements can be found in the following documentation:

### References

Operating Manual Universal/Turning/Milling/Grinding, chapter "Collision avoidance" > "Creating a kinematic structure" > "Kinematic elements"

### Setting up parameters

Set the following machine data to set up the kinematics transformation technology:

| MD18866 $MN_MM_NUM_KIN_TRAFOS | Maximum number of NC transformation objects |
|---|---|
| ≥ 1 | This machine data specifies the maximum number of NC transformations that can be defined with kinematic chains (variable). |

## Setting up system variables

Use the following system variables to set up the kinematics transformation:

| System variable | Meaning |
|---|---|
| $NT_NAME[n] | Name of the transformation that appears on the user interface |
| $NT_TRAFO_TYPE[n] | Transformation type for turning and milling machines: TRAORI_STAT |
| $NT_T_CHAIN_LAST_ELEM[n] | Name of the chain element that describes the end of the chain in the direction of the tool |
| $NT_P_CHAIN_LAST_ELEM[n] | Name of the chain element that describes the end of the chain in the direction of the workpiece |
| $NT_GEO_AX_NAME[n, 0 - 2] | Names of the chain elements that describe the 1st to 3rd GEO axis (X, Y, Z) |
| $NT_ROT_AX_NAME[n, 0 - 2] | Name of the chain element that describes the 1st to 3rd rotary axes (rotary indexing table, swivel axis, tool spindle) |
| $NT_IDENT[n, 0 - 2] | Settings to adapt the screen form |
| $NT_AUX_POS[n, 0 - 2] | Values for retraction in the direction of 1st to 3rd GEO axis (X, Y, Z) [1] |
| $NT_CNTRL[n] | Settings to adapt to the machine |

n: Number of the transformation data set with n = 1, 2, 3, ... (MD18866 $MN_MM_NUM_KIN_TRAFOS)

[1] See manufacturer cycle CUST_800.SPF markers _M47 and _M48

## Meaning of the decimal places of $NT_IDENT[n, 0 - 2]

You can set the following display variants of the input screens for turning on milling machines:

| Decimal place | Meaning | |
|---|---|---|
| ONES | Selection of the alignment | |
| | 0 = | Beta/gamma |
| | 1 = | Beta/gamma + direct alignment (B/SP) |
| THOUSANDS | Selection field: +/- direction reference of the swivel axis | |
| | 0 = | No selection on the user interface, always minus |
| | 4 = | +/- selection via user interface, default: Minus |
| | 5 = | No selection on the user interface, always plus |
| | 9 = | +/- selection via user interface, default: Plus |
| TEN THOUSANDS | Selection box: Correct tool tip when swiveling | |
| | 2 = | Do not list toggle |
| | 3 = | List toggle |
| ONE MILLION | Selection box: Retraction mode 0X and | |
| TEN MILLION | Selection box: Retraction mode X0 (the permissible retraction modes result from the sum of the individual values: 0 ... 15) | |
| | 0 = | No retraction |
| | 1 = | Retraction Z [2] |
| | 2 = | Retraction Z, XY [2] |
| | 4 = | Maximum retraction in tool direction |
| | 8 = | Incremental retraction in tool direction |

| Decimal place | Meaning | |
|---|---|---|
| HUNDRED MILLION | Enable data set | |
| | 4 = | Kinematics active (fixed value) |

[2] See system variables $NT_AUX_POS[n,0] to $NT_AUX_POS[n,2]

### Meaning of the bits of $NT_CNTRL[n]

You can make the following settings for the machine adaptation:

| | |
|---|---|
| Bit 0 = 1 | Fixed sequence of axes<br>For easier handling of the data, the sequence of the axes is fixed:<br>1. Rotary table<br>2. Swivel axis<br>3. Tool spindle |
| Bit 1 = 1 | Rotary table is spindle |
| Bit 2 = 0 | Swivel axis is not a spindle |
| Bit 3 = 0 | Tool spindle is not a spindle (during turning) |
| Bit 4 = 0 | 1st rotary axis has no Hirth joint (rotary table) [3] |
| Bit 5 = 0 | 2nd rotary axis has no Hirth joint (swivel axis) [3] |
| Bit 6 = 0 | 3rd rotary axis has no Hirth joint (tool spindle) [3] |

[3] It is not permissible that either the 2nd or the 3rd rotary axis has a Hirth joint.

## 18.8.3 Behavior at reset and power on

### Behavior at reset

Make the following settings so that the kinematics transformation is retained after reset and the turning tool can be used in JOG mode:

| MD20110 $MC_RESET_MODE_MASK | | Definition of the basic control setting after reset / part program end |
|---|---|---|
| Bit 7 = 1 | The current setting for active transformation is retained after reset / part program end. | |
| Bit 16 = 1 | The current setting of the master spindle is retained. | |

| MD20112 $MC_START_MODE_MASK | Definition of the basic control setting after part program start |
|---|---|
| Bit 7 = 0 | Start behavior, "active kinematics transformation" |

| MD20152 $MC_GCODE_RESET_MODE[ ] | Reset behavior of G groups |
|---|---|
| [5] = 1 | The current setting of the 6th G group is also retained after reset / part program end (G17, ...). |

| | |
|---|---|
| [14] = 1 | The current setting of the 15th G group is also retained after reset / part program end (G94, ...). |
| [28] = 1 | The current setting of the 29th G group is also retained after reset / part program end (DIAMxxx). |

### Behavior at power on

In order to retain the kinematics transformation after a power on, you must also make the following additional settings:

| MD20144 $MC_TRAFO_MODE_MASK | Selection of the kinematics transformation function |
|---|---|
| Bit 1 = 1 | Reselection of the transformation after power-up |

| MD24008 $MC_CHSFRAME_POWERON_MASK | Reset system frames after power on |
|---|---|
| Bit 4 = 0 | Do not delete workpiece reference points after power on |

**Note**

The CUTMODK function cannot be retained after power on.

## 18.8.4 Setting the tool clamping angle

If the clamped turning tool is not in the G18 plane at the spindle zero position, the clamping angle must be set.

For this, the position of the tool spindle must be entered as a clamping angle at which the tool is mechanically in the desired position.

### Procedure




1. Select the "Parameter" operating area.




2. Press the "Tool list" softkey.
   The "Tool List" window opens.
3. Select a suitable turning tool in the list.




4. Press the "Additional data" softkey.
   The "Additional Data - ..." window opens.
   The "Additional data" softkey is only active if a turning tool for which the "Additional Data" window is configured is selected.
5. Enter the required angle in the "Angle" input field.

**Note**

The clamping angle is stored in the $TC_DPROT cutting edge parameter.

## 18.8.5 Setting up the Hirth joint

If the tool spindle or the swivel axis has a Hirth joint with, for example, incrementing of 2.5° (360°/2.5° = 144 division), you can set up the following:

### Settings

| MD30505 $MA_HIRTH_IS_ACTIVE | Axis is indexing axis with Hirth joint |
|---|---|
| = 0 | Tool spindle / swivel axis is not a real Hirth axis |

| MD30330 $MA MODULO RANGE | Size of the modulo range |
|---|---|
| = 360° | Effective for modulo axis |

| MD30340 $MA_MODULO_RANGE_START | Start position of the modulo range. |
|---|---|
| = 0° | |

| MD30501 $MA_INDEX_AX_NUMERATOR | Indexing axis, numerator for equidistant positions |
|---|---|
| = 360° | Effective for non-modulo axis |

| MD30502 $MA_INDEX_AX_DENOMINATOR | Indexing axis, denominator for equidistant positions |
|---|---|
| = 144 | Defines the value of the denominator for calculation of the distances between two indexing positions for equidistant positions. For modulo axes, this is therefore the number of indexing positions. |

### Swivel axis has Hirth joint

| $NT_CNTRL[n] | |
|---|---|
| Bit 5 = 1 | Swivel axis has a Hirth joint<br>[n] transformation name |

### Tool spindle has Hirth joint

| $NT_CNTRL[n] | |
|---|---|
| Bit 6 = 1 | Tool spindle has a Hirth joint<br>[n] transformation name |

## 18.8.6 Different coordinate systems for milling and turning

In boring mills, the rotary table usually rotates around the Y axis (B axis). During turning, the workpiece must rotate around the Z axis (C axis) in the G18 plane. Therefore, different coordinate systems must be used on this machine for milling and turning.

### Settings

Set the appropriate coordinate system via the following channel-specific machine data (values, for example, for boring mills):

| MD52000 $MCS_DISP_COORDINATE_SYSTEM | Position of the coordinate system (variable) |
|---|---|
| = 16 | Y up, Z to the right |

| MD52001 DISP_COORDINATE_SYSTEM_2 | Coordinate system for turning on milling machines (variable) |
|---|---|
| = 0 | Z up, X to the right |

Describe the rotation of the coordinate system as follows at the _M70 marker in the CUST_800.SPF cycle:

```
_M70:          ; Rotation of the coordinate system for turning
N800703 $P_WPFRAME=$P_WPFRAME:CROT(Y,-90):CROT(X,-90)
```

## 18.8.7 Adaptations for CUST_800

Adapt these markers in the CUST_800.SPF cycle for turning on milling machines:

### Function markers

```
_M40:                          ; User Init
N800400 SETMS(1)               ; Sets tool spindle as main spindle
N800401 DIAMOF                 ; No transverse axis defined as diameter
                               programming for milling
GOTOF _MEND

_M47:                          ; Retract axis Z to MCS fixed position
                               before tool alignment
```

```
GOTOF _MEND

_M48:                              ; Retract axis Z and then the XY axes to
                                   MCS fixed position before tool alignment
GOTOF _MEND

_M70:                              ; Alignment tool: Prepare
IF($P_SMODE[1]==1)                 ; Spindle mode of tool spindle
N800700 SPOS[1]=0                  ; Tool spindle as rotary axis C
ENDIF
N800701 SETMS(2)                   ; Sets table as main spindle
N800702 DIAM90                     ; Transverse axis defined as diameter
                                   programming for turning
GOTO _MEND

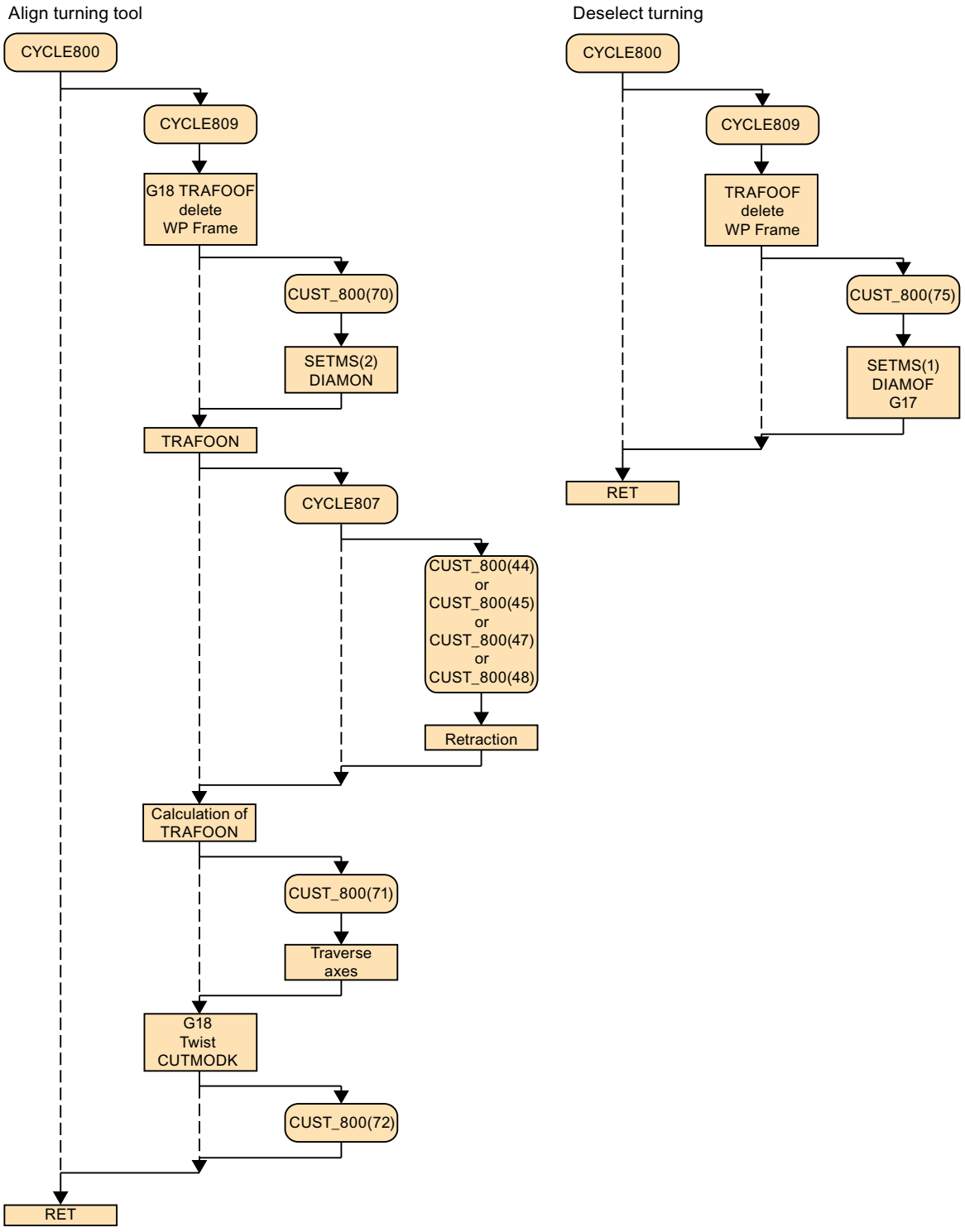
_M71:                              ; Alignment tool: Swivel rotary axis
_AX1=$PC_TRAFO_ROT_CHAN_AX_IN[1]   ; Rotary axis B
_AX2=$PC_TRAFO_ROT_CHAN_AX_IN[2]   ; Tool spindle
IF(_TC_N_WZ==0)                    ; No tooltip tracking
N800710 TRAFOOF
ENDIF
SBLON
N800711 G0 G40 G60 G90 AX[_AX1]=_A1 AX[_AX2]=_A2
SBLOF
GOTOF _MEND

_M72:                              ; Alignment tool: End
GOTOF _MEND

_M75:                              ; Alignment tool: Deselection
N800750 G17                        ; Standard plane for milling
N800751 SETMS(1)                   ; Sets tool spindle as main spindle
N800752 DIAMOF                     ; No transverse axis defined as diameter
                                   programming for milling
GOTOF _MEND
```

## Simplified flow diagrams

During tool retraction, the markers _M44 to _M48 are run through before the rotary axis positioning in accordance with the retraction variant.



For swiveling with retraction to MCS fixed position, see Chapter "Manufacturer cycle CUST_800.SPF (Page 435)".

## 18.8.8 Rotary axis positioning during block search

After a block search, you must position the rotary axis as follows (refer to swivel plane). Generally, there are two rotary axes (swivel axis and tool spindle):

### Procedure

Read out the position value of the rotary axis via the following system variable:

| System variable | Meaning |
|---|---|
| $AC_RETPOINT[axis] | Repositioning point in the ASUB |

You position the rotary axis in the CYCPE_MA.SPF user cycle.

**Note**

So that the old rotary axis position is not swiveled in, start the block search in the G code program after the block for the turning tool alignment.

For further information on CYCPE_MA.SPF, see Chapter "Extending the PROG_EVENT standard cycle (Page 358)".

If you have switched off the kinematics transformation, e.g. for a tool change, you can switch the transformation on again with the following program extension:

```
DEF STRING[32] U_TRAFO_NAME
U_TRAFO_NAME=$AC_TRAFO_NAME
TRAFOOF
...
TRAFOON(U_TRAFO_NAME)
```

## 18.8.9 Follow-up for rotary axis positioning after block search

The rotary axes must be positioned after a block search. If the positioning is done with follow-up, it is possible that the linear axes will reach the software limit switch. To prevent this, the positioning of the rotary axes is always done without follow-up by default.

### Supplement for CYCPE1MA.SPF

If you nevertheless want to position the rotary axes with follow-up, you can supplement the following in the user cycle CYCPE1MA.SPF for example:

```
; Position rotary axes "with follow-up":
IF($P_PROG_EVENT==5) ; Block search
```

```
  IF($PC_TRAFO_TYPE_NAME=="TRAORI_STAT")
    IF(($NT_IDENT[($P_TRAFO_PARSET MOD 1000),0] _DEC5)>=2)
      _TC_N_WZ=1 ; Correct tool
      CUST_800(71,,$AC_RETPOINT[$PC_TRAFO_ROT_CHAN_AX_IN[1]],
      $AC_RETPOINT[$PC_TRAFO_ROT_CHAN_AX_IN[2]])
    ENDIF
  ENDIF
ENDIF
```

# 18.9 High-speed machining

## 18.9.1 High speed settings: Configuring CYCLE832

### Function

The High Speed Settings function (CYCLE832) is used to preset data for the machining of sculptured surfaces so that optimum machining is possible. The call of CYCLE832 contains three parameters:

- Tolerance
- Machining type (technology)
- Input of the orientation tolerance
  The 3rd parameter is significant only for machines with multi-axis orientation transformation (TRAORI). By default, it is assigned the value 1.



**Software option**

To use this function, you need the software option:

"Advanced Surface"

### Tolerance / orientation tolerance

The tolerance value for the linear axes (geometry axes) is transferred to the CNC with the NC command CTOL. CTOL = root(3) * tolerance value.

Tolerance value = 0: Tolerance value (CTOL) is loaded from the following setting data:

| | |
|---|---|
| SD55446 $SCS_MILL_TOL_VALUE_ROUGH | Tolerance value for roughing CYCLE832 |
| SD55447 $SCS_MILL_TOL_VALUE_SEMIFIN | Tolerance value for semi-finishing CYCLE832 |
| SD55448 $SCS_MILL_TOL_VALUE_FINISH | Tolerance value for finishing CYCLE832 |

Example: CYCLE832(0,_ROUGH,1) → tolerance value from SD55446 is used.

If rotary axes are involved in the machining (multi-axis transformation), the tolerance value is transferred to the CNC with the NC command OTOL.

There are 2 options available for this:

1. Direct entry of the orientation tolerance OTOL
   Machining types: _ORI_ROUGH, _ORI_SEMIFIN, _ORI_FINISH
   To enter the orientation tolerance in the input screen for CYCLE832, bit 1 =1 must be set in the
   setting data SD55220 $SCS_FUNCTION_MASK_MILL_TOL_SET
2. Tolerance is transferred with a factor on the NC command OTOL
   Machining types: _ROUGH, _SEMIFIN, _FINISH
   This factor can be set in the following channel-specific setting data for each machining type:

| | |
|---|---|
| SD55441 $SCS_MILL_TOL_FACTOR_ROUGH | Factor, tolerance of the rotary axes for roughing of the G group 59. |
| SD55442 $SCS_MILL_TOL_FACTOR_SEMIFIN | Factor, tolerance of the rotary axes for roughing-finishing of the G group 59. |
| SD55443 $SCS_MILL_TOL_FACTOR_FINISH | Factor, tolerance of the rotary axes for finishing of the G group 59. |
| = 10 | |

For machines for which the orientation tolerance is not significant, the cycle setting data SD55441 to 55443 can be set to a value of 0. No orientation tolerance is then calculated in CYCLE832 (OTOL= -1).

## Machining type and technology G group 59

The machining types of the technology G group 59 are permanently assigned in CYCLE832 or in CUST_832.SPF:

| Machining type | Plain text | Technology G group 59 | Field index |
|---|---|---|---|
| Deselection | _OFF | DYNNORM | 0 |
| Roughing | _ROUGH | DYNROUGH | 2 |
| Rough-finishing | _SEMIFIN | DYNSEMIFIN | 3 |
| Finishing | _FINISH | DYNFINISH | 4 |
| Roughing *) | _ORI_ROUGH | DYNROUGH | 2 |
| Semi-finishing *) | _ORI_SEMIFIN | DYNSEMIFIN | 3 |
| Finishing *) | _ORI_FINISH | DYNFINISH | 4 |

*) With entry of an orientation tolerance

Dynamic parameters can be adapted to the respective machining operation with the technology G groups. Using the commands of technology G group 59, the value of the following channel and axis-specific machine data is activated using the corresponding **array index**:

| | |
|---|---|
| MD20600 $MC_MAX_PATH_JERK[0..4] | Path dependent maximum jerk. |
| MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL[0..4] | Influence of path curvature on path dynamic response. |
| MD20603 $MC_CURV_EFFECT_ON_PATH_JERK[0..4] | Influence of path curvature on path jerk. |
| MD32300 $MA_MAX_AX_ACCEL[0..4] | Maximum axis acceleration. |

| | |
|---|---|
| MD32310 $MA_MAX_ACCEL_OVL_FACTOR[0..4] | Overload factor for axial velocity jumps. |
| MD32431 $MA_ MAX_AX_JERK[0..4] | Maximum axial jerk for path motion. |
| MD32432 $MA_PATH_TRANS_JERK_LIM[0..4] | Maximum axial jerk at the block transition in continuous-path mode. |

**Note**

During the optimization of the machine axes, the values (note array index) of the specified machine data must be set correctly.

### Machining type, deselection

When CYCLE832 is deselected, the G groups are programmed for the settings during the program run time; these settings are declared in machine data MD20150 $MC_GCODE_RESET_VALUES[ ]. This involves the G groups 1, 4, 10, 12, 21, 24, 30, 45, 59, 61.

The contour and orientation tolerance is deactivated (CTOL=-1, OTOL=-1).

## 18.9.2 Manufacturer cycle CUST_832.SPF

### Function

In contrast to the settings (G functions) by CYCLE832.SPF, these settings can be modified in the manufacturer cycle CUST_832.SPF.

The following markers are prepared in CUST_832.SPF:

| | |
|---|---|
| `_M_NORM:` | For deselection _OFF CYCLE832 |
| `_M_FINISH:` | For finishing _FINISH and _ORI_FINISH |
| `_M_SEMIFINISH:` | For semi-finishing _SEMIFIN and _ORI_SEMIFIN |
| `_M_ROUGH:` | For roughing _ROUGH and _ORI_ROUGH |

The following G functions are programmed in CUST_832.SPF:

```
SOFT
COMPCAD
G645
FIFOCTRL
UPATH
;FFWON
```

Dependent on the machining type:

```
DYNNORM
DYNFINISH
DYNSEMIFIN
DYNROUGH
```

The programming of FGREF () is useful when machining with active multi-axis transformation (e.g. TRAORI). In this case, in CUST_832.SPF, variable _FGREF is pre-assigned a value of 10 mm. This value can also be modified. In CYCLE832.SPF, the value of variable _FGREF is written to the rotary axes involved in the machining, which are declared as orientation axis of a 5-axis transformation, using the FGREF(rotary axis) command. When G70/G700 is active, the value from _FGREF is converted into inches before writing to the command FGREF.

## Take into account more than three rotary axes (orientation axes) in CYCLE832

In CYCLE832, a maximum of three rotary axes of the orientation transformation (TRAORI) are taken into account for FGREF. If more than three rotary axes are declared per channel for the orientation transformation, you can write the value to FGREF using the following syntax in the CUST_832:

```
FGREF[AA]=$AA_FGREF[C]
```

C = rotary axis 1 (axis is taken into account by CYCLE832)
AA = rotary axis 4

In the manufacturer cycle CUST_832.SPF, parameter _MOTOL is available for the evaluation of the "Multi-axis programming yes/no".

| | |
|---|---|
| `_MOTOL=0` | No multi-axis programming |
| `_MOTOL=1` | Multi-axis programming where the orientation tolerance is entered |

With the following setting data, additional G functions can be programmed in CUST_832.SPF for using multi-axis programming:

| SD55220 $SCS_FUNCTION_MASK_MILL_TOL_SET | | Function screen form, high speed settings CYCLE832 |
|---|---|---|
| Bit 1 | Orientation tolerance (OTOL) as a factor of the tolerance (CTOL) or directly entered | |

## Example

Using _MOTOL in CUST_832.SPF for the marker for finishing:

```
_M_FINISH:
IF _MOTOL==1            ; if multi-axis programming
    ORISON
ELSE                    ; no multi-axis programming
    ORISOF
ENDIF
...
```

## 18.9.3 Top Surface: CYCLE832 extension

### Function

Besides Advanced Surface, the High Speed Settings (CYCLE832) function also offers the extension Top Surface. You can use Top Surface to improve the quality of free-form surfaces. If set actively, the continuous-path control is optimized within contour tolerances. The CYCLE832 call also contains the smoothing parameter.

**Software option**

To use this function, you need the software option:

"Top Surface"

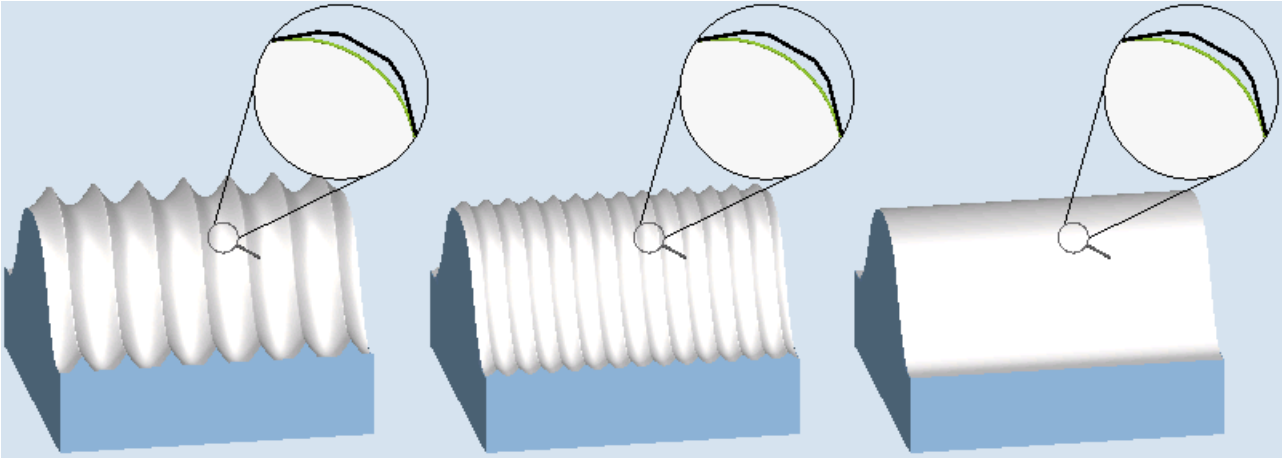


Figure 18-16 Roughing, semi-finishing, or finishing with active smoothing

### Tolerance / orientation tolerance

The tolerance value for Top Surface is transferred to the NC as a contour tolerance with command CTOL and as a multiple axis transformation with command OTOL.

### Setting up parameters

Set the following machine and setting data to set up the Top Surface extension:

| MD28071 $MC_MM_NUM_SURF_LEVELS | Dimensioning the COMPSURF function (DRAM) |
|---|---|
| = 6 | Defines the size of the internal data structures of the COMPSURF function. With larger values, the attainable surface quality increases, but so do the computing time and memory requirements. |

| SD55220 $SCS_FUNC-TION_MASK_MILL_TOL_SET | Function screen form, high speed settings CY-CLE832 |
|---|---|
| Bit 2 | Mold making function can be selected in High-Speed Settings screen form |

| | |
|---|---|
| = 0 | The best available mold making function is automatically used:<br>• Top Surface not active → Advanced Surface<br>• Top Surface active → Top Surface |
| = 1 | The mold making function (Advanced Surface or Top Surface) can be selected in the input screenform. |
| Bit 4 | Display selection "Smoothing" |
| = 0 | Do not display selection "Smoothing" in the screen form (corresponds to smoothing = yes) |
| = 1 | Display selection "Smoothing" in the screen form |

### Extension in manufacturer cycle CUST_832.SPF

The following labels are provided in manufacturing cycle CUST_832.spf for Top Surface:

| | |
|---|---|
| `31 _M_TOP_SURFACE_FINISH:` | Finishing with COMPSURF |
| `32 _M_TOP_SURFACE_SEMIFINISH:` | Semi-finishing with COMPSURF |
| `33 _M_TOP_SURFACE_ROUGH:` | Roughing with COMPSURF |

The following G commands are programmed in labels 31 to 33:

```
SOFT
COMPSURF
G645
FIFOCTRL
UPATH
;FFWON
```

Depending on the machining type used, the following G commands of G group 59 are programmed:

```
DYNFINISH
DYNSEMIFIN
DYNROUGH
```

## 18.10 Measuring cycles and measurement functions

### 18.10.1 Measuring cycles and measurement functions, general

### Requirement

Two types of electronic probes are used for measuring:

- Probe to measure the workpiece
- Probe to measure the tool

The electronic probe must be connected to the NCU 7x0.

The electronic probe is called a measuring input in the following document.

## References

NCU Manual: Digital inputs / outputs

## Setting the switching behavior

You set the electrical polarity of the connected probe using the following general machine data:

| MD13200 $MN_MEAS_PROBE_LOW_ACTIVE | Polarity change of the probe |
|---|---|
| = 0 | Probe in the non-deflected state, 0 V default value<br>Probe in the deflected state, 24 V |
| = 1 | Probe in the non-deflected state, 24 V<br>Probe in the deflected state, 0 V |

| MD13210 $MN_MEAS_TYPE | Measuring type for distributed drives |
|---|---|
| = 0 | Default value |

## Testing the probe function

You can test the switching function of the probe by manually deflecting it and checking the following PLC interface signals:

| **DB 10** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| DBB107 | | | | | | | Probe actuated | |
| | | | | | | | Probe 2 | Probe 1 |

To test the switching behavior and the measured value transfer, use an NC test program with, for example, the following NC commands:

| | |
|---|---|
| MEAS | Measurement with deletion of distance-to-go |
| $AC_MEA[n] | Check the switching operation, n = measurement input number |
| $AA_MW[axis name] | Measured value of the axes in workpiece coordinates |
| $AA_MM[axis name] | Measured value of the axes in machine coordinates |

## Example of a test program

| **Program code** | **Comment** |
|---|---|
| %_N_PRUEF_MESSTASTER_MPF | |
| $PATH=/_N_MPF_DIR | ;Testing program probe connection |
| N00 DEF INT MTSIGNAL | ;Bit memory to check the switching state |
| N05 G17 G54 T="3D_Taster" D1 | ;Select tool geometry for probe |
| N10 M06 | ;Activate tool |
| N15 G0 G90 X0 F150 | ;Starting position and meas. velocity |
| N20 MEAS=1 G1 X100 | ;Measurement at measuring input 1 in the X axis |

| Program code | Comment |
|---|---|
| N30 MTSIGNAL=$AC_MEA[1] | ;Switching operation at the 1st measuring input completed, ;YES / NO |
| N35 IF MTSIGNAL == 0 GOTOF _FEHL1 | ;Evaluation of the signal |
| N40 R1=$AA_MM[X] | ;Save measured value in machine coordinates at R1 |
| N45 R2=$AA_MW[X] | ;Save measured value in workpiece coordinates at R2 |
| N50 M0 | ;Check measured value in R1/R2 |
| N55 M30 | |
| N60 _FEHL1: MSG ("Probe does not switch!") | |
| N65 M0 | |
| N70 M30 | |

## General function settings for measuring cycles

| MD51740 $MNS_MEA_FUNCTION_MASK | Measuring cycles function screen |
|---|---|
| Bit 0 | Workpiece, monitoring the calibration status |
| Bit 1 | Workpiece, length reference of the probe in the infeed axis with relation to the sphere circumference |
| = 0 | Tool length L1, referred to the center of the probe sphere |
| = 1 | Tool length L1, referred to the sphere circumference of the probe sphere |
| Bit 2 | Workpiece, offset into tool, tool carrier taken into account |
| Bit 4 | Workpiece, probe permanently affixed to the machine (not in a spindle) |
| Bit 5 | Workpiece, coarse offset in ZO if no fine ZO has been set up |
| Bit 6 | Measuring blocks in the measuring cycles with feed override 100% |
| Bit 15 | Workpiece, traverse measuring block with path behavior G60 |
| Bit 16 | Tool, turning technology, consideration of tool holders |

**Note**

After changing this machine data, the probe must be re-calibrated.

| MD52740 $MCS_MEA_FUNCTION_MASK | Measuring cycles function screen |
|---|---|
| Bit 0 | Workpiece probe connected to measurement sensing input 2 |
| = 0 | Workpiece probe at the CNC measuring input 1, active - default |
| = 1 | Workpiece probe at the CNC measuring input 2, active |
| Bit 1 | Workpiece, turning technology, measuring in the third geometry axis (Y) |
| Bit 2 | Calculation of the correction angle with/without active orientation transformation |
| Bit 16 | Tool probe connected to measurement sensing input 2 |
| = 0 | Tool probe at the CNC measuring input 1, active |
| = 1 | Workpiece probe at the CNC measuring input 2, active - default |
| Bit 17 | Tool measurement turning, recoding of cutting edge position |

| MD54740 $SNS_MEA_FUNCTION_MASK | | Measuring cycles function screen |
|---|---|---|
| Bit 0 | Workpiece, repeat measurement if TSA, TDIF exceeded | |
| Bit 1 | Workpiece, repeat measurement with M0 if TSA, TDIF exceeded | |
| Bit 2 | Workpiece, stop at M0 if TUL, TLL exceeded | |
| Bit 3 | Workpiece, accept calibrated probe sphere radius in tool data | |
| Bit 4 | Workpiece, plane can vary between calibration and measurement | |
| Bit 16 | Tool, repeat measurement if TDIF, TSA exceeded | |
| Bit 17 | Tool, repeat measurement with stop at M0 if TDIF, TSA exceeded | |
| Bit 19 | Tool, milling, reduction of spindle speed during last probing | |

| MD55740 $SCS_MEA_FUNCTION_MASK | | Measuring cycles function screen |
|---|---|---|
| Bit 0 | Workpiece, collision monitoring for intermediate positioning | |
| Bit 1 | Workpiece, coupling of spindle position with coordinate rotation around the infeed axis | |
| Bit 2 | Workpiece, reverse positioning direction if spindle position is coupled | |
| Bit 3 | Workpiece, no repetition if probe does not switch | |
| Bit 4 | Workpiece, 1st probing for every measurement with rapid measuring feedrate | |
| Bit 5 | Workpiece, rapid retraction following measurement | |
| Bit 6 | Workpiece, deactivate probe during spindle positioning | |
| Bit 7 | Kinematics, scaling on the basis of the input values of the rotary axis vectors | |
| Bit 8 | Kinematics, measuring with swiveling (TCARR) or active TRAORI | |
| Bit 9 | Kinematics, user-specific scaling of the rotary axis vectors | |
| Bit 10 | Kinematics, measure completely with reference | |
| Bit 11 | Kinematics, number of measurement points when measuring calibration sphere | |
| Bit 15 | Workpiece, calibrating radius with starting point in center of ring, JOG mode only | |
| Bit 16 | Tool, collision monitoring for intermediate positioning | |
| Bit 17 | Tool, no repetition if probe does not switch | |
| Bit 19 | Tool, rapid retraction following measurement | |
| Bit 28 | Continuous measuring standard log in JOG (attach) | |
| Bit 29 | Table format for measuring standard log in JOG | |

## 18.10.2 Manufacturer and user cycle CUST_MEACYC.SPF

### Function

The CUST_MEACYC.SPF is part of the measuring cycle functionality. It is called in every measuring cycle before and after executing the measurement task. The CUST_MEACYC.SPF acts in the same way when measuring in the JOG mode and measuring in the AUTOMATIC mode.

You can use the CUST_MEACYC.SPF to program and execute sequences that are necessary before and/or after a measurement (e.g. activating/deactivating a probe).

**Note**

Copy the CUST_MEACYC cycle before performing the changes from the directory NC data/Cycles/Standard cycles to the directory NC data/Cycles/Manufacture cycles or NC data/Cycles/User cycles. Changes for CUST_MEACYC only take effect in these directories.

## CUST_MEACYC structogram

Sequence in the AUTOMATIC mode, using measure bore as example, CYCLE977 with probe type 712.

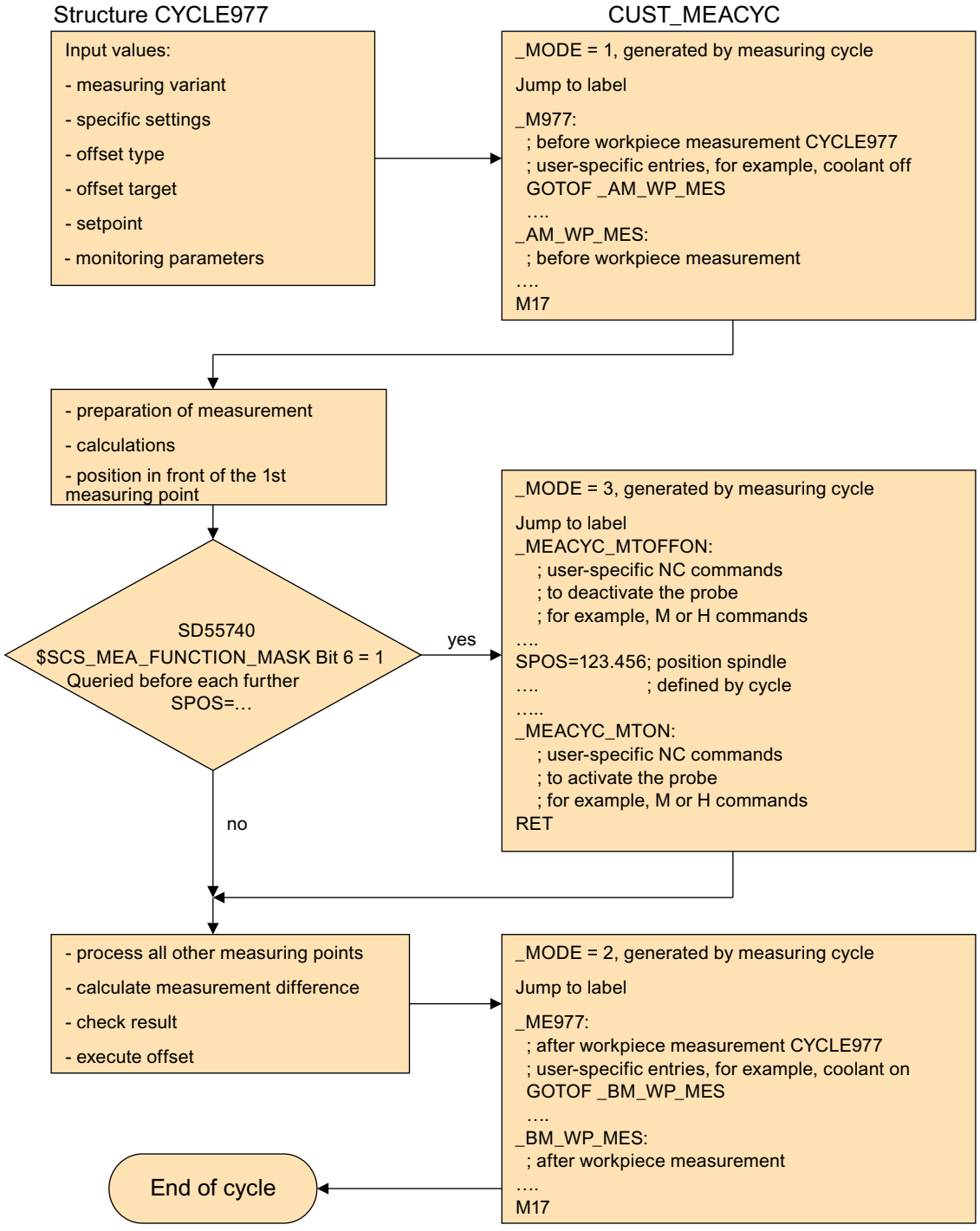



Figure 18-17 Structure: CYCLE977 / CUST_MEACYC

**Note**

**Commenting out spindle SPOS commands**

If the spindle SPOS command is to be commented out in the CUST_MEACYC.SPF cycle, set the cycle setting in SD55740 $SCS_MEA_FUNCTION_MASK to active with Bit 1 = 1.

You can see how to comment out a spindle SPOS command in the CUST_MEACYC.SPF cycle in the following example:

### Example

```
;
;*#1 Example application for spindle pre-positioning if there is no coupling
;*##1 between spindle and the coordinate system active!
;*#1 This example can be removed if necessary!
;*#2 User example for spindel positioning, if you do not have a coupling
;*##2 between the spindel and the coordinate system!
;*#2 If this lines are not necessary, the user can delete this example!
;
; cycle is calling from the AUTOMATIC-mode AND the spindel is positionable
IF NOT(_OVI[14]) AND NOT(_SPOS_MOD)
  ; no coupling between the spindel and the coordinate system!
  ; SD55740, Bit 1 = 0
  IF ($SCS_MEA_FUNCTION_MASK B_AND 'B10')==0
  ; SPOS=0 ; <===== commented out
  ENDIF
ENDIF
M17
;
```

## 18.10.3 Measuring in JOG mode

### Requirements

You have already made the settings from the previous Chapter "Measuring cycles and measurement functions, general (Page 463)".

## Workpiece measurement

**For milling:**

- The probe has been inserted in the tool spindle.
- The probe has been selected in the tool list as type 7xx (probe).
- The probe is activated as tool in the current NC channel.

## Tool measurement

To measure tools, an appropriate probe must be located in the machine space so that this can be reliably and safely reached with a tool in the spindle.

The following tool types are supported with measure tool:

- Milling technology: Tool types 1xx and 2xx
- Turning technology: Tool types 5xx, 1xx and 2xx

For the specified tool types, the tool lengths and the tool radii can be measured.

### Setting general data

| SD54782 $SNS_J_MEA_FUNCTION_MASK_TOOL | Settings for the input screen, measuring in JOG, tool measurement. |
|---|---|
| Bit 2 = 1 | Activates the "Enable automatic tool measuring" function. |
| Bit 3 = 1 | Selects the tool probe calibration data field, enable. |
| Bit 10 = 1 | Selects the individual cutting edge measurement |
| Bit 11 = 1 | Selects the spindle reversal when calibrating in the plane |

| MD51784 $MNS_J_MEA_T_PROBE_APPR_AX_DIR [i] | |
|---|---|
| Measuring in JOG: Approach direction in the plane to the tool probe | |
| -2 | Negative direction, 2nd measuring axis |
| -1 | Negative direction, 1st measuring axis (default setting) |
| 0 | Positive direction, 1st measuring axis |
| 1 | |
| 2 | Positive direction, 2nd measuring axis |

| MD52750 $MCS_J_MEA_FIXPOINT | Z value for measurement with respect to a fixed point |
|---|---|
| MD52751 $MCS_J_MEA_MAGN_GLAS_POS | Zoom-in position for tool measurement |
| [0] | Position in the 1st axis |
| [1] | Position in the 2nd axis |

## Checking and setting the general machine data

| MD11450 $MN_SEARCH_RUN_MODE | "Search run" programming. |
|---|---|
| Bit 1 = 1 | Automatic ASUB start after output of the action blocks (see also MD11620 $MN_PROG_EVENT_NAME). Alarm 10208 is not output until the ASUB is completed. |

| MD11604 $MN_ASUP_START_PRIO_LEVEL | Priorities from which ASUP_START_MASK is effective. |
|---|---|
| = 1 - 64 | Priorities for ASUP_START_MASK. |

Setting channel-specific machine data

| MD20050 $MC_AXCONF_GEOAX_ASIGN_TAB[ ] | Assignment, geometry axis to channel axis. |
|---|---|
| [0] | For measuring in JOG mode, all of the geometry axes must be available. Preferably XYZ. |
| [1] | |
| [2] | |

| MD20110 $MC_RESET_MODE_MASK | Defines the basic control settings after reset / TP end, |
|---|---|
| = 4045H | Minimum value |
| Bit 0 = 1 | Basic control setting after power on and reset. |
| Bit 2 = 1 | |
| Bit 6 = 1 | |
| Bit 14 = 1 | |

| MD20112 $MC_START_MODE_MASK | Definition of the basic control setting after part program start. |
|---|---|
| = 400H | Minimum value |
| Bit 6 = 0 | Definition of the basic control setting after NC start. |

| MD20310 $MC_TOOL_MANAGEMENT_MASK | Activates tool manager functions |
|---|---|
| = 4002H | Minimum value |
| Bit 1 = 1 | Tool manager and monitoring functions active |
| Bit 14 = 1 | Automatic tool change for RESET and Start. |

| MD28082 $MC_SYSTEM_FRAME_MASK | System frames (SRAM). |
|---|---|
| Bit 0 = 0 | Do not setup system frames for actual value setting and scratching. |
| Bit 0 = 1 | Setup system frames for actual value setting and scratching. |

The following channel-specific machine data is only effective if the system frame actual value setting and scratching is set up using the above mentioned machine data MD28082 $MC_SYSTEM_FRAME_MASK, bit 0 = 0.

| MD24006 $MC_CHSFRAME_RESET_MASK | Active system frames after reset. |
|---|---|
| Bit 0 = 1 | Reset response of the system frames or actual value setting. |

| MD24007 $MC_CHSFRAME_RESET_CLEAR_MASK | Clear system frames at reset. |
|---|---|
| Bit 0 = 0 | Clear response of the system frames on reset. |

**Note**

If you create the conditions described in this section and you have set and checked the machine setting data, then in JOG mode, you can measure a workpiece using a workpiece probe at a milling machine!

In JOG mode, you can measure a tool using a tool probe at a milling machine or lathe!

A description is provided in the following chapters as to which settings you can make in order to adapt measuring to the specific requirements of your particular machine.

### 18.10.3.1 Measuring workpieces at the milling machines

Measuring in the "Machine" operating area can be appropriately adapted to the specific requirements using the following channel-specific general machine data and channel-specific cycle setting data.

#### General configuration machine data

| SD55630 $SCS_MEA_FEED_MEASURE | Measuring feedrate for workpiece measurement in JOG and AUTOMATIC |
|---|---|
| 300 | Default value |

**Note**

**Measuring feedrate for workpiece measuring**

All measuring cycles use the value saved in SD54611 as the measuring feedrate value following calibration of the workpiece probe. A different measuring feedrate can be assigned for each calibration field [n].

When calibrating the probe, either the measuring feedrate from SD55630 $SCS_MEA_FEED_MEASURE is used, or the measuring feedrate can be overwritten in the input screen when calibrating. Bit 4 = 1 must be set in the SD54760 $SNS_MEA_FUNCTION_MASK_PIECE for this purpose.

| MD52207 $MCS_AXIS_USAGE_ATTRIB[n] | Attributes of the axes<br>[n] channel axis number |
|---|---|
| Bit 6 | Request rotational axis as offset target for measuring |
| = 0 | Default value |
| = 1 | Rotary axis is displayed as offset target in the "Measuring in JOG" window |
| Bit 9 | SPOS behavior of main spindle influenced in connection with measuring cycles |
| = 0 | The positioning behavior of the tool spindle is not influenced |
| = 1 | The measuring cycles do not position the tool spindle (no SPOS) |

**Note**

If the tool spindle is never SPOS-capable, Bit 9 = 1 must be set.

| MD51751 $MNS_J_MEA_M_DIST_MANUELL | Measurement path in mm, in front of and behind the measurement point |
|---|---|
| = 10 | Default value |

| MD51757 $MNS_J_MEA_COLL_MONIT_FEED | Position feedrate in mm/min, in the working plane for active collision monitoring |
|---|---|
| = 1000 | Default value |

| MD51758 $MNS_J_MEA_COLL_MONIT_POS_FEED | Position feedrate in mm/min, in the infeed axis for active collision monitoring |
|---|---|
| = 1000 | Default value |

| MD51770 $MNS_J_MEA_CAL_RING_DIAM[n] | Pre-assignment of the calibration ring diameter in mm specifically for the calibration data sets |
|---|---|
| = -1 | Default value |

| MD51772 $MNS_J_MEA_CAL_HEIGHT_FEEDAX[n] | Pre-assignment of the calibration height in mm in the infeed axis, specifically for the calibration data sets |
|---|---|
| = -99999 | Default value |

## General cycle setting data

| SD54780 $SNS_J_MEA_FUNCTION_MASK_PIECE | Configuration of the input screens for measuring in the "Machine" operating area in the JOG operating mode |
|---|---|
| = 512 | Default value |
| Bit 2 = 1 | Activates measurements with an electronic probe |

| | |
|---|---|
| Bit 3 = 1 | Selects the probe calibration data, enable |
| Bit 6 = 1 | Selects ZO correction in the basis reference (SETFRAME), enable |
| Bit 7 = 1 | Selects ZO correction in channel-specific basic frames, enable |
| Bit 8 = 1 | Selects ZO correction in the global basic frame, enable |
| Bit 9 = 1 | Selects ZO correction in adjustable frames, enable |
| Bit 16 = 1 | Selects workpiece measurement with spindle reversal |
| Bit 17 = 1 | Selects align workpiece probe in the switching direction |
| Bit 28 = 1 | Selects adjust workpiece length, enable |

### 18.10.3.2 Measuring tools at the milling machines

#### Measuring feedrate for tool measurement in JOG and AUTOMATIC

| SD55628 $SCS_MEA_TP_FEED_MEASURE | | Calibrate measuring feedrate for tool probe and measure tool with stationary spindle. |
|---|---|---|
| = 300 | Default value | |

**Note**

**Measuring feedrate for tool measuring**

All measuring cycles use the value saved in SD54636 or SD54651 as the measuring feedrate after the tool probe has been calibrated. A different measuring feedrate can be assigned for each calibration field [n].

When calibrating the probe, either the measuring feedrate from SD55628 SCS_MEA_TP_FEED_MEASURE is used, or the measuring feedrate can be overwritten in the input screen when calibrating. To do this, SD54762 $SNS_MEA_FUNCTION_MASK_TOOL bit 4 must be set to 1.

In the following setting data, index [k] stands for the number of the current data field (probe number -1) of the probe.

| SD54633 $SNS_MEA_TP_TYPE[k] | Probe type, cube/disk |
|---|---|
| = 0 | Compatibility (measuring cycles: cube, turning surface shows probe, milling surface shows disk) |
| = 101 | Disk in XY, working plane G17 |
| = 201 | Disk in ZX, working plane G18 |
| = 301 | Disk in YZ, working plane G19 |
| = 2 | Probe |
| = 3 | Cube |

The following setting data is used to define in which axes and directions it is possible to calibrate a tool probe.

| SD54632 $SNS_MEA_T_PROBE_ALLOW_AX_DIR[k] | | Axes and directions for "calibration" |
|---|---|---|
| = 133 | Default value | |

| Decimal place | | |
|---|---|---|
| ONES | 1st axis | |
| | = 0 | axis not possible |
| | = 1 | only minus direction |
| | = 2 | only plus direction |
| | = 3 | both directions |
| TENS | 2nd axis | |
| | = 0 | axis not possible |
| | = 1 | only minus direction |
| | = 2 | only plus direction |
| | = 3 | both directions |
| HUNDREDS | 3rd axis | |
| | = 0 | axis not possible |
| | = 1 | only minus direction |
| | = 2 | only plus direction |
| | = 3 | both directions |

## Example

If the general setting data SD54632 $SNS_MEA_T_PROBE_ALLOW_AX_DIR[k] has the value 123, the tool probe is calibrated as follows in the G17 plane:

- X in both directions
- Y only in plus direction
- Z only in minus direction

| SD54631 $MNS_MEA_T_PROBE_DIAM_LENGTH[k] | | Effective diameter of the tool probe for length measurement. |
|---|---|---|
| = 0 | Default value | |

## General cycle machine/setting data for tool probe

| MD51780 $MNS_J_MEA_T_PROBE_DIAM_RAD[k] | | Effective diameter of the tool probe for radius measurement. |
|---|---|---|
| = 0 | Default value | |

| SD54634 $SNS_MEA_TP_CAL_MEASURE_DEPTH[n] | | Distance between the tool probe upper edge and the milling tool lower edge. |
|---|---|---|
| = 2 | Default value | |

| MD51752 $MNS_J_MEA_M_DIST_TOOL_LENGTH | | Measuring path to measure the tool length. |
|---|---|---|
| = 2 | Default value | |

| MD51753 $MNS_J_MEA_M_DIST_TOOL_RADIUS | | Measuring path to measure the tool radius. |
|---|---|---|
| = 1 | Default value | |

| MD51786 $MNS_J_MEA_T_PROBE_MEASURE_DIST | | Measuring path to calibrate the probe or for measurements with stationary spindle. |
|---|---|---|
| = 10 | Default value | |

## General cycle setting data when measuring with rotating spindle

| SD54670 $SNS_MEA_CM_MAX_PERI_SPEED[0] | | Maximum permissible peripheral speed of the tool to be measured. |
|---|---|---|
| = 100 | Default value | |

| SD54671 $SNS_MEA_CM_MAX_REVOLUTIONS[0] | | Maximum permissible tool speed of the tool to be measured. The speed is automatically reduced when exceeded. |
|---|---|---|
| = 1000 | Default value | |

| SD54672 $SNS_MEA_CM_MAX_FEEDRATE[0] | | Maximum permissible feedrate to probe the tool to be measured at the probe. |
|---|---|---|
| = 20 | Default value | |

| SD54673 SNS_MEA_CM_MIN_FEEDRATE[0] | | Minimum feedrate for the first probing of the tool to be measured at the probe. This avoids excessively small feedrates for large tool radii. |
|---|---|---|
| = 1 | Default value | |

| SD54674 $SNS_MEA_CM_SPIND_ROT_DIR[0] | | Spindle direction of rotation to measure tools. |
|---|---|---|
| = 4 | Spindle rotation the same as M4 (default value) | |

**Note**

If the spindle is already rotating when the measuring cycle is called, this direction of rotation remains independent of the setting of this data.

| SD54675 $SNS_MEA_CM_FEEDFACTOR_1[0] | Feedrate factor 1 |
|---|---|
| = 10 | Default value |
| = 0 | Only single probing with the feedrate calculated by the cycle. However, as a minimum, the value from SD54673[0] $SNS_MEA_CM_MIN_FEEDRATE. |
| >= 1 | First probing with feedrate. However, as a minimum with the value from SD54673[0] $SNS_MEA_CM_MIN_FEEDRATE)<br>SD54675[0] $SNS_MEA_CM_FEEDFACTOR_1 |

| SD54676 $SNS_MEA_CM_FEEDFACTOR_2[0] | Feedrate factor 2 |
|---|---|
| = 0 | Second probing with the feedrate calculated by the measuring cycle. This is only effective for SD54673[0] $SNS_MEA_CM_FEEDFACTOR_1 > 0, default value |
| >= 1 | Second probing with the calculated feedrate from SD54673[0] $SNS_MEA_CM_MIN_FEEDRATE feedrate factor 2.<br>Third probing with the calculated feedrate. |

**Note**

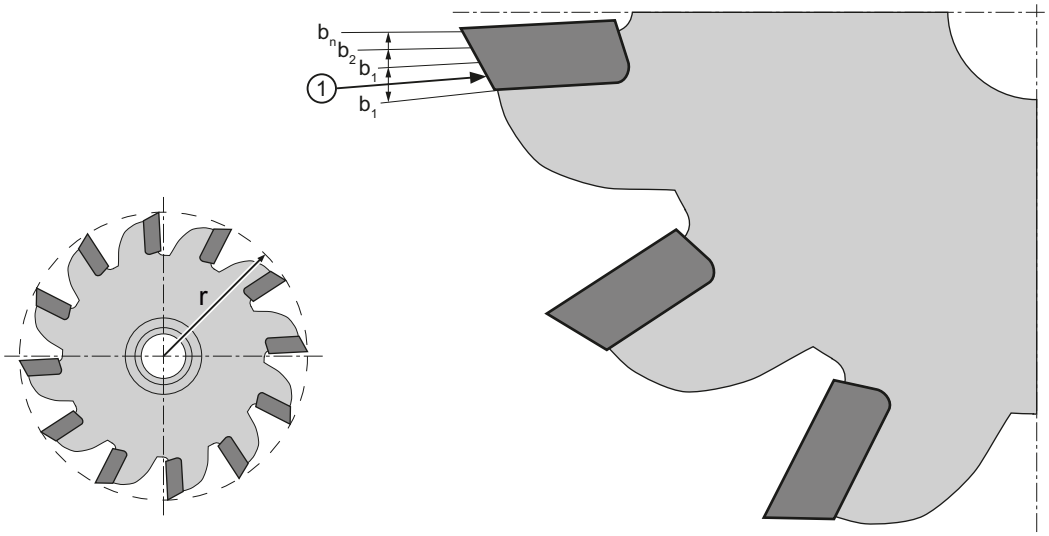
Feedrate factor 2 should be less than feedrate factor 1.

| SD54677 $SNS_MEA_CM_MEASURING_ACCURACY[0] | Specified measuring accuracy.<br>The value of this parameter always refers to the last probing of the tool at the probe! |
|---|---|
| = 0.005 | Default value |

| SD54692 $SNS_MEA_T_CIRCULAR_ARC_DIST | Arc distance for the single cutting edge measurement of the radius. |
|---|---|
| = 0.25 *) | Default value |

| SD54693 $SNS_MEA_T_MAX_STEPS | Maximum number of probe contacts for single cutting edge measurement of the radius. |
|---|---|
| = 10 *) | Default value |

*) If the value = 0, then the "highest point on the longest cutting edge" is not sought, but the measured value of the contact with the rotating spindle taken.




r Effective radius of the tool

b Arc distance = SD54692

n Maximum number of probe contacts = SD54693

① 1st measuring point.

Figure 18-18 Single cutting edge measurement

## General cycle setting data for correction using the correction tables when forming measurements with rotating spindle.

| SD54691 $SNS_MEA_T_PROBE_OFFSET | Activate the measuring result compensation |
|---|---|
| = 0 | No data, default value |
| = 1 | Correction in the cycle. This is only effective if SD54689 $SNS_MEA_T_PROBE_MANUFACTURER>0. |
| = 2 | Correction using user-defined correction table |

| SD54689 $SNS_MEA_T_PROBE_MANUFACTURER | Activates pre-configured compensation tables for several tool probe models (customer-specific) |
|---|---|
| = 0 | No data, default value |
| = 1 | TT130 (Heidenhain) |
| = 2 | TS27R (Renishaw) |

### 18.10.3.3 Measuring tools at the turning machines

#### Measuring feedrate for tool measurement in turning operation

| SD55628 $SCS_MEA_TP_FEED_MEASURE | | Calibrate measuring feedrate for tool probe and measure tool with stationary spindle. |
|---|---|---|
| = 300 | Default value | |

**Note**

**Measuring feedrate for tool measuring**

All measuring cycles use the value saved in SD54636 or SD54651 as the measuring feedrate after the tool probe has been calibrated. A different measuring feedrate can be assigned for each calibration field [n].

When calibrating the probe, either the measuring feedrate from SD55628 is used, or the measuring feedrate can be overwritten in the input screen when calibrating. To do this, SD54762 $SNS_MEA_FUNCTION_MASK_TOOL bit 4 must be set to 1.

#### General cycle machine/setting data for measuring path/feedrate

| SD42950 $SC_TOOL_LENGTH_TYP | | Assignment of the tool length offset independent of tool type. |
|---|---|---|
| = 0 | Measuring turning tools, type 5xx (default value) | |
| = 2 | Measuring turning tools, type 5xx, drilling and milling tools, type 1xx, 2xx | |

| MD51786 $MNS_J_MEA_T_PROBE_MEASURE_DIST | | Measuring path to calibrate the probe or for measurements with stationary spindle. |
|---|---|---|
| = 10 | Default value | |

| SD55628 $SCS_MEA_TP_FEED_MEASURE | | Feedrate for the calibration of a tool probe with stationary spindle. |
|---|---|---|
| = 300 | Default value | |

#### Calibrating tool probes

Use the following general cycle machine data to set the access level as of which the "Calibrate probe" softkey is displayed. This requires that the automatic tool measurement is enabled using the setting data SD54782 (bit 2 = 1).

| MD51070 $MNS_ACCESS_CAL_TOOL_PROBE | | Calibrate tool probe access level. |
|---|---|---|
| = 7 | Access rights: Access level 7 (keyswitch 0) | |

| | |
|---|---|
| = 4 | Read access rights: Access level 4 (keyswitch 3), default value |
| = 3 | Write access: Access level 3 (user) |

### 18.10.3.4 Measuring without electronic probe in JOG

#### Measure workpiece

For manual measuring functions in the JOG operating mode, for which a rotating spindle is required, the following applies:

- The channel state must be reset at the instant in time that the measured value is transferred.
- The spindle can be moved or positioned either using the T,S,M menu or using the bits in DB3x.DBB30.
- The following preconditions apply when using the DB3x.DBB30 technology functions with fixed speed:

| Setting the machine data | | |
|---|---|---|
| MD10709 $MN_PROG_SD_POWERON_INIT_TAB [0] = 0 | | Setting data to be initialized |
| MD35035 $MA_SPIND_FUNCTION_MASK | Bit 4 = 0<br>Bit 5 = 1 | Spindle functions |
| SD43200 $SA_SPIND_S | | Speed setting value |

**Note**

The entry in SD43200 $SA_SPIND_S is kept at power on.

With these settings, the spindle velocity for manual operation, which is set in SD41200 $SN_JOG_SPIND_SET_VELO, is no longer effective.

## 18.10.4 Measuring in the AUTOMATIC mode

#### Requirement



**Software option**

In order to use the "Measuring in AUTOMATIC" function, you require the software option: "Measuring cycles"

You have already made the settings from Chapter "Measuring cycles and measurement functions, general (Page 463)".

## Workpiece measurement

**Sequence in milling technology:**

1. The probe has been selected in the tool list as type 7xx (probe).
2. The probe has been inserted in the tool spindle.
3. The probe is activated in the actual NC channel.

**Sequence in turning technology:**

1. Tool type 580 (3D probe, turning) is selected.
2. Tool is activated in the actual NC channel.

## Tool measurement

To measure tools, an appropriate probe must be located in the machine space so that this can be reliably and safely reached with a tool in the spindle.

The following tool types are supported with tool measurement:

- Milling technology: Tool types 1xx and 2xx
- Turning technology: Tool types 5xx, 1xx and 2xx

For the specified tool types, the tool lengths and the tool radii can be measured.

## Channel-specific cycle setting data

Using the following channel-specific cycle setting data, you can adapt the **workpiece and tool measuring** in the "Program" operating area to specific requirements. As a general rule, no changes are required.

| SD55613 $SCS_MEA_RESULT_DISPLAY | | Selects the screen display of the measuring result |
|---|---|---|
| = 0 | No screen display of the measuring result (default value). | |
| = 1 | Screen display of the measuring result is displayed for 8 seconds. | |
| = 3 | The measuring cycle stops an internal machine data, the measuring result is statically displayed on the screen!<br>Continue with NC start, the measuring result screen is deselected. | |
| = 4 | The measuring result is only displayed on the screen for cycle alarms 61303, 61304, 61305, 61306.<br>Continue with NC start, the measuring result display on the screen is deselected. | |

| SD55614 $SCS_MEA_RESULT_MRD | Setting of the program control of the measurement result display MRD |
|---|---|
| = 0 | Switch off the measurement result display |
| = 1 | Switch on the measurement result display |

| SD55623 $SCS_MEA_EMPIRIC_VALUE[n] | Empirical values |
|---|---|
| = 0 | Default value |

| SD55618 $SCS_MEA_SIM_ENABLE | Measuring cycle simulation |
|---|---|
| = 0 | When calling measuring cycles in conjunction with SINUMERIK Operate, ShopMill or ShopTurn simulation, the measuring cycles are skipped, default value. |
| = 1 | Default setting:<br>When calling measuring cycles in conjunction with SINUMERIK Operate, ShopMill or ShopTurn simulation, the measuring cycles are run through. However, no corrections are performed and there is no logging. There is no screen display of the measuring result. |

For further information on SD55618, please refer to:

### References

Programming Manual, Measuring Cycles; Section "Description" > "Behavior on block search, dry run, program testing, simulation"

### Configuring input screens for measuring cycles in the program editor

Using the following setting data, you configure the display screen and the correction options when making measurements in the AUTOMATIC mode:

- SD54760 $SNS_MEA_FUNCTION_MASK_PIECE measure workpiece: Measuring cycles in the milling area, see Chapter "Measuring workpieces, general (Page 482)"
- SD54762 $SNS_MEA_FUNCTION_MASK_TOOL measure tool: Measuring cycles in the milling area, see Chapter "Measuring tools at the milling machines (Page 486)"

**Note**

If you have created the prerequisites described in this chapter and you have set and checked the machine/setting data, you can perform measurements at the machine with a workpiece or tool probe in the AUTOMATIC mode!

If you wish to set up workpiece measurement on a turning machine, then additional settings are required. Refer to Chapter "Measuring workpieces at the turning machines (Page 492)" for the settings.

If you wish to set-up tool measurement using a tool probe, then you must also make additional settings. Please refer to Chapters "Measuring tools at the turning machines (Page 493)" or "Measuring tools at the milling machines (Page 486)" for the settings.

A description is provided in the following chapters as to which settings you can make in order to adapt measuring to the specific requirements of your particular machine.

## 18.10.4.1 Measuring workpieces, general

Using the following general and channel-specific cycle setting data, you can appropriately adapt workpiece measurement to the specific requirements in the "Program" operating area.

## Requirement

You have already made the settings from the Chapter "Measuring cycles and measurement functions, general (Page 463)".

## Settings

General cycle setting data

| SD54760 $SNS_MEA_FUNCTION_MASK_PIECE | | Settings for the input screen, measuring cycles in AUTOMATIC, workpiece measurement. |
|---|---|---|
| Bit 1 | Display select softkey 3D measurement | |
| Bit 3 | Enable probe calibration field to be selected | |
| Bit 4 | Select calibration input measuring feed [1)] | |
| Bit 6 | Enable NP correction in the basis reference (SETFRAME) to be selected | |
| Bit 7 | Enable NP correction in channel-specific basic frame to be selected | |
| Bit 8 | Enable NP correction in global basic frame to be selected | |
| Bit 9 | Enable NP correction in adjustable frame to be selected | |
| Bit 10 | Enable NP correction coarse and fine to be selected | |
| Bit 11 | Select tool offset, geometry and wear | |
| Bit 12 | Select tool offset, not inverted and inverted | |
| Bit 13 | Select tool offset L1, R or L1, L2, L3 R | |
| Bit 14 | Select tool offset, zero offset (_TZL) | |
| Bit 15 | Select tool offset, dimensional difference monitoring (_TDIF) | |
| Bit 16 | Select workpiece measurement with spindle reversal | |
| Bit 17 | Selects align workpiece probe in the switching direction | |
| Bit 18 | Select number of measurements (_NMSP) | |
| Bit 19 | Select offset with mean value generation (_TMV) [1)] | |
| Bit 20 | Select experience values (_EVNUM) | |
| Bit 21 | Select total setting-up offset | |
| Bit 22 | Select calibration to unknown or to known center point | |
| Bit 24 | Select calibration with/without position deviation | |
| Bit 25 | Select zero offset when measuring the angulation of the spindle | |
| Bit 26 | Selects tool offset, do not enable | |
| Bit 27 | Selects measure tolerance of the linear vector for kinematics, do not enable | |
| Bit 28 | Enable tool offset adjust length to be selected | |

Channel-specific cycle setting data

| SD55630 $SCS_MEA_FEED_MEASURE | | Measuring feedrate [mm/rev] |
|---|---|---|
| = 300 | Measuring feedrate when calibrating the workpiece probe | |

**Note**

**Measuring feedrate for workpiece measuring**

All measuring cycles use the value saved in SD54611 $SNS_MEA_WP_FEED[n] as measuring feedrate after the tool probe has been calibrated.

A different measuring feedrate can be assigned for each calibration field [n].

When calibrating the probe, either the measuring feedrate from SD55630 $SCS_MEA_FEED__MEASURE is used, or the measuring feedrate can be overwritten in the input screen when calibrating. To do this, SD54760 MEA_FUNCTION_MASK_PIECE bit 4 must be set to 1.

| SD55632 $SCS_MEA_FEED_RAPID_IN_PERCENT | Percentage reduction |
|---|---|
| = 50 | Default value of the percentage reduction of the rapid traverse velocity for internal cycle intermediate positioning without collision monitoring.<br>The setting of the channel-specific cycle setting data SD55600 SCS_MEA_COLLISION_MONITORING must be = 0. |

| SD55634 $SCS_MEA_FEED_PLANE_VALUE | Feedrate of the intermediate positioning in the working plane with active collision monitoring. |
|---|---|
| = 1000 | Default value |

| SD55636 $SCS_MEA_FEED_FEEDAX_VALUE | Feedrate of the intermediate positioning in the infeed axis with active collision monitoring. |
|---|---|
| = 1000 | Default value |

| SD55638 $SCS_MEA_FEED_FAST_MEASURE | Fast measuring feedrate |
|---|---|
| = 900 | Default value |

| SD55640 $SCS_MEA_FEED_CIRCLE | Feedrate with circle programming |
|---|---|
| = 1000 | Default value |

| SD55642 $SCS_MEA_EDGE_SAVE_ANG | Safety angle when measuring a corner (CYCLE961) |
|---|---|
| = 0 | Degrees |

This angle is added to the angle that the user entered between the workpiece reference edge and the 1st axis of the active coordinate system in a positive direction, and refers to the corner measurement in the automatic mode (CYCLE961).

The angle of the user specifications is designated "α0" in the parameterization screen corner measurement. For compatibility programs, the designation is _STA1.

Notice:

For compatibility programs, $SCS_MEA_EDGE_SAVE_ANG should be set = 10.

### 18.10.4.2 Measuring workpieces at the milling machines

Measuring in the "Program" operating area can be adapted corresponding to the specific requirements using the channel-specific cycle setting data.

## Settings

| SD55625 $SCS_MEA_AVERAGE_VALUE[n] | Number of mean values. |
|---|---|
| = 0 | Default value |

The following cycle setting data are written to by the measuring cycles with the "Calibrate probe" measuring function. User parameterization is not necessary here. However, after the probe has been calibrated, you can check these values and if required, evaluate the probe quality, e.g. for position deviations, no values > 0.1 mm should be reached. Otherwise, the probe must be mechanically readjusted.

The measuring feedrate at the time of calibration is used for all subsequent applications of the measuring cycles.

**Note**

Observe the manufacturer's instructions for the probe.

| | |
|---|---|
| SD54600 $SNS_MEA_WP_BALL_DIAM[n] | Effective diameter of the probe ball of the workpiece probe. |
| SD54601 $SNS_MEA_WP_TRIG_MINUS_DIR_AX1[n] | Trigger point, minus direction, 1st measuring axis in the plane. |
| SD54602 $SNS_MEA_WP_TRIG_PLUS_DIR_AX1[n] | Trigger point, plus direction, 1st measuring axis in the plane. |
| SD54603 $SNS_MEA_WP_TRIG_MINUS_DIR_AX2[n] | Trigger point, minus direction, 2nd measuring axis in the plane. |
| SD54604 $SNS_MEA_WP_TRIG_PLUS_DIR_AX2[n] | Trigger point, plus direction, 2nd measuring axis in the plane. |
| SD54605 $SNS_MEA_WP_TRIG_MINUS_DIR_AX3[n] | Trigger point, minus direction, 3rd measuring axis in the tool direction. |

| | |
|---|---|
| SD54606 $SNS_MEA_WP_TRIG_PLUS_DIR_AX3[n] | Trigger point, plus direction,<br>3rd measuring axis opposite to the tool direction. In the default case = 0. |
| SD54607 $SNS_MEA_WP_POS_DEV_AX1[n] | Position deviation,<br>1st measuring axis in the plane. |
| SD54608 $SNS_MEA_WP_POS_DEV_AX2[n] | Position deviation,<br>2nd measuring axis in the plane. |
| SD54609 $SNS_MEA_WP_STATUS_RT | Calibration status of the axis positions |
| SD54610 $SNS_MEA_WP_STATUS_GEN[n] | Calibration status |
| SD54611 $SNS_MEA_WP_FEED[n] | Measuring feedrate when calibrating |

### 18.10.4.3 Measuring tools at the milling machines

## Settings

| SD54762 $SNS_MEA_FUNCTION_MASK_TOOL | Setting for the input screen, measuring cycles in the automatic mode, tool measurement |
|---|---|
| Bit 3 | Setting for the input screen, measuring cycles in the automatic mode, tool measurement, enable tool probe calibration data field to be selected |
| Bit 4 | Select calibration input measuring feed (VMS) [1] |
| Bit 5 | Select input feedrate and spindle speeds when probing |
| Bit 7 | Select measurement in MCS and WCS |
| Bit 8 | Select measurement, absolute and incremental |
| Bit 9 | Select tool offset, geometry and wear |
| Bit 10 | Select individual cutting edge measurement |
| Bit 11 | Selects spindle reversal when calibrating in the plane |
| Bit 12 | Select number of measurements (_NMSP) |
| Bit 13 | Select experience values (_EVNUM) |

[1] Input, measurement feedrate is valid for AUTOMATIC and JOG

## Calibration data of the tool probe, referred to the machine coordinate system

Before calibration is started, the position of the tool probe in the machine coordinate system (MCS) must be entered into the following general cycle setting data. In this case, the reference point is the outer diameter or the tool length of the active tool in the spindle. If there is no tool in the spindle, the reference points are the spindle center point and the tool reference point at the spindle.

**Note**

**Calibrate probe**

If you have calibrated the tool probe in JOG mode, then the calibration data has already been correctly entered in: SD54632 $SNS_MEA_TP_AX_DIR_AUTO_CAL[k]

You do not need to recalibrate the tool probe in the AUTOMATIC mode.

Index [k] stands for the number of the actual data field (probe number -1).

| | |
|---|---|
| SD54625 $SNS_MEA_TP_TRIG_MINUS_DIR_AX1[k] | Trigger point of the 1st measuring axis in the negative direction. |
| SD54626 $SNS_MEA_TP_TRIG_PLUS_DIR_AX1[k] | Trigger point of the 1st measuring axis in the positive direction. |
| SD54627 $SNS_MEA_TP_TRIG_MINUS_DIR_AX2[k] | Trigger point of the 2nd measuring axis in the negative direction. |
| SD54628 $SNS_MEA_TP_TRIG_PLUS_DIR_AX2[k] | Trigger point of the 2nd measuring axis in the positive direction. |
| SD54629 $SNS_MEA_TP_TRIG_MINUS_DIR_AX3[k] | Trigger point of the 3rd measuring axis in the negative direction. |
| SD54630 $SNS_MEA_TP_TRIG_PLUS_DIR_AX3[k] | Trigger point of the 3rd measuring axis in the positive direction. |
| SD54631 $SNS_MEA_TP_EDGE_DISK_SIZE[k] | Tool probe, edge length / disk diameter. |
| SD54632 $SNS_MEA_TP_AX_DIR_AUTO_CAL[k] | Axes and directions for calibrating in AUTOMATIC mode. |
| SD54634 $SNS_MEA_TP_CAL_MEASURE_DEPTH[k] | Distance between the upper edge of the tool probe and lower edge of the tool (calibration depth, measuring depth for milling radius). |
| SD54635 $SNS_MEA_TPW_STATUS_GEN[k] | Calibration status |
| SD54636 $SNS_MEA_TPW_FEED[k] | Measuring feedrate when calibrating |

The general cycle setting data SD54632 $SNS_MEA_TP_AX_DIR_AUTO_CAL, is used to define in which axes and directions it is possible to calibrate the tool probe.

| Decimal place | | |
|---|---|---|
| ONES | 1st axis | |
| | = 0 | axis not possible |
| | = 1 | only minus direction |
| | = 2 | only plus direction |
| | = 3 | both directions |
| TENS | 2nd axis | |

| Decimal place | | |
|---|---|---|
| | = 0 | axis not possible |
| | = 1 | only minus direction |
| | = 2 | only plus direction |
| | = 3 | both directions |
| HUNDREDS | 3rd axis | |
| | = 0 | axis not possible |
| | = 1 | only minus direction |
| | = 2 | only plus direction |
| | = 3 | both directions |

## Example

If the general cycle machine data SD54632 $SNS_MEA_TP_AX_DIR_AUTO_CAL has the value 123, the tool probe is calibrated as follows in the G17 plane:

- X in both directions
- Y only in plus direction
- Z only in minus direction

| SD54633 $SNS_MEA_TP_TYPE[k] | Probe version |
|---|---|
| = 0 | Compatibility (measuring cycles: cube, turning surface shows probe, milling surface shows disk) |
| = 101 | Disk in XY, working plane G17. |
| = 201 | Disk in ZX, working plane G18. |
| = 301 | Disk in YZ, working plane G19. |
| = 2 | Probe |
| = 3 | Cube |

## Calibration data of the tool probe referred to the workpiece coordinate system

Before calibration is started, the position of the tool probe in the workpiece coordinate system (WCS) must be roughly entered into the following general cycle setting data. In this case, the reference point is the outer diameter or the tool length of the active tool in the spindle. If there is no tool in the spindle, the reference points are the spindle center point and the tool reference point at the spindle.

**Note**

When measuring tools, ensure that the data of the adjustable zero offset or the basic reference always correspond to the data when calibrating (measuring in WCS!).

Always make measurements and calibrate with the same adjustable zero offset.

| | |
|---|---|
| SD54640 $SNS_MEA_TPW_TRIG_MINUS_DIR_AX1[k] | Trigger point of the 1st measuring axis in the negative direction. |
| SD54641 $SNS_MEA_TPW_TRIG_PLUS_DIR_AX1[k] | Trigger point of the 1st measuring axis in the positive direction. |
| SD54642 $SNS_MEA_TPW_TRIG_MINUS_DIR_AX2[k] | Trigger point of the 2nd measuring axis in the negative direction. |
| SD54643 $SNS_MEA_TPW_TRIG_PLUS_DIR_AX2[k] | Trigger point of the 2nd measuring axis in the positive direction. |
| SD54644 $SNS_MEA_TPW_TRIG_MINUS_DIR_AX3[k] | Trigger point of the 3rd measuring axis in the negative direction. |
| SD54645 $SNS_MEA_TPW_TRIG_PLUS_DIR_AX3[k] | Trigger point of the 3rd measuring axis in the positive direction. |
| SD54646 $SNS_MEA_TPW_EDGE_DISK_SIZE[k] | Tool probe, edge length / disk diameter. |
| SD54647 $SNS_MEA_TPW_AX_DIR_AUTO_CAL[k] | Automatic calibration of tool probe, enable axes/directions |
| SD54648 $SNS_MEA_TPW_TYPE[k] | Probe version |
| = 0 | Compatibility (measuring cycles: cube, turning surface shows probe, milling surface shows disk) |
| = 101 | Disk in XY, working plane G17. |
| = 201 | Disk in ZX, working plane G18. |
| = 301 | Disk in YZ, working plane G19. |
| = 2 | Probe |
| = 3 | Cube |
| SD54649 $SNS_MEA_TPW_CAL_MEASURE_DEPTH[k] | Distance between the upper edge of the tool probe and lower edge of the tool (calibration depth, measuring depth for milling radius). |
| SD54650 $SNS_MEA_TPW_STATUS_GEN[k] | Calibration status |
| SD54651 $SNS_MEA_TPW_FEED[k] | Measuring feedrate when calibrating |

The following general cycle setting data SD54647 $SNS_MEA_TPW_AX_DIR_AUTO_CAL is used to define in which axes and directions it is possible to calibrate a tool probe.

| **Decimal place** | | |
|---|---|---|
| ONES | 1st axis | |
| | = 0 | axis not possible |
| | = 1 | only minus direction |
| | = 2 | only plus direction |
| | = 3 | both directions |
| TENS | 2nd axis | |
| | = 0 | axis not possible |
| | = 1 | only minus direction |
| | = 2 | only plus direction |
| | = 3 | both directions |
| HUNDREDS | 3rd axis | |
| | = 0 | axis not possible |
| | = 1 | only minus direction |

| Decimal place | | |
|---|---|---|
| | = 2 | only plus direction |
| | = 3 | both directions |

## Example

If the general cycle machine data SD54647 $SNS_MEA_TPW_AX_DIR_AUTO_CAL has the value 123, the tool probe is calibrated as follows in the G17 plane:

- X in both directions
- Y only in plus direction
- Z only in minus direction

## Monitoring when measuring with a rotating spindle

| SD54670 $SNS_MEA_CM_MAX_PERI_SPEED[0] | | Maximum permissible peripheral speed of the tool to be measured. |
|---|---|---|
| = 100 | Default value | |

| SD54671 $SNS_MEA_CM_MAX_REVOLUTIONS[0] | | Maximum permissible tool speed of the tool to be measured. The speed is automatically reduced when exceeded. |
|---|---|---|
| = 1000 | Default value | |

| SD54672 $SNS_MEA_CM_MAX_FEEDRATE[0] | | Maximum permissible feedrate to probe the tool to be measured at the probe. |
|---|---|---|
| = 20 | Default value | |

| SD54673 $SNS_MEA_CM_MIN_FEEDRATE[0] | | Minimum feedrate for the first probing of the tool to be measured at the probe. This avoids excessively small feedrates for large tool radii. |
|---|---|---|
| = 1 | Default value | |

| SD54674 $SNS_MEA_CM_SPIND_ROT_DIR[0] | | Spindle direction of rotation to measure tools. |
|---|---|---|
| 4 = M4 | Default value | |

**Note**

If the spindle is already rotating when the measuring cycle is called, this direction of rotation remains independent of the setting of this data.

| SD54675 $SNS_MEA_CM_FEEDFACTOR_1[0] | Feedrate factor 1 |
|---|---|
| = 10 | Default value |
| = 0 | Only single probing with the feedrate calculated by the cycle. However, as a minimum, the value from SD54673[0] $SNS_MEA_CM_MIN_FEEDRATE. |
| = ≥1 | First probing with feedrate. However, as a minimum with the value from SD54673[0] $SNS_MEA_CM_MIN_FEEDRATE)<br>SD54675[0] $SNS_MEA_CM_FEEDFACTOR_1 |

| SD54676 $SNS_MEA_CM_FEEDFACTOR_2[0] | Feedrate factor 2 |
|---|---|
| = 0 | Second probing with the feedrate calculated by the cycle. This is only effective for SD54673 $SNS_MEA_CM_FEEDFACTOR_1[0] > 0, default value. |
| = ≥1 | Second probing with the calculated feedrate from SD54673 $SNS_MEA_CM_MIN_FEEDRATE[0] feedrate factor 2.<br>Third probing with the calculated feedrate. |

**Note**

Feedrate factor 2 should be less than feedrate factor 1.

| SD54677 $SNS_MEA_CM_MEASURING_ACCURACY[0] | Specified measuring accuracy. The value of this parameter always refers to the last probing of the tool at the probe. |
|---|---|
| = 0.005 | Default value |

## Measurement with rotating spindle: Measured value correction using correction tables

| SD54691 $SNS_MEA_T_PROBE_OFFSET | Activates the measuring result correction |
|---|---|
| = 0 | No data, default value |
| = 1 | Correction in the cycle. This is only effective if SD54690 $SNS_MEA_T_PROBE_MANUFACTURER>0. |
| = 2 | Correction using user-defined correction table |

| SD54689 $SNS_MEA_T_PROBE_MANUFACTURER | Activate pre-configured compensation tables for several tool probe models (customer-specific). |
|---|---|
| = 0 | No data, default value |
| = 1 | TT130 (Heidenhain) |
| = 2 | TS27R (Renishaw) |

### Correction values for users

If the general cycle setting data SD54691 $SNS_MEA_T_PROBE_OFFSET= 2, the following settings apply:

| | | |
|---|---|---|
| SD54695 to SD54700 | Correction values for radius measurement. | See the subsequent general cycle setting data. |
| SD54705 to SD54710 | Correction values for length measurement. | See the subsequent general cycle setting data. |

| SD54695 $SNS_MEA_RESULT_OFFSET_TAB_RAD1[n]<br>SD54705 $SNS_MEA_RESULT_OFFSET_TAB_LEN1[n] | Radius measurement<br>Length measurement |
|---|---|
| = 0 | 0 |
| = 1 | 1st radius |
| = 2 | 2nd radius |
| = 3 | 3rd radius |
| = 4 | 4th radius |

## 18.10.4.4 Measuring workpieces at the turning machines

### General settings for turning

| MD52740 $MNS_MEA_FUNCTION_MASK | Measuring cycles function screen |
|---|---|
| Bit 1 | Workpiece measurement, measuring in the third geometry axis (Y, when turning |
| = 0 | The measuring cycles do not support a third geometry axis (Y axis)! |
| =1 | Setpoint input and parameterization (SETVAL, _TUL, _TLL, SZO) refer to the third geometry axis (Y axis). The correction of the tool length or zero offset is however realized in the second geometry axis (X axis, ordinate) active components (i.e. measure in Y and correct in X). The correction target can be influenced using parameter _KNUM! |
| Bit 16 | Measuring input, tool probe |

### Calibration data of the tool probe, referred to the machine coordinate system

Before calibration is started, the position of the tool probe in the machine coordinate system (MCS) must be entered into the following general cycle setting data.

| | |
|---|---|
| SD54615 $SNS_MEA_CAL_EDGE_BASE_AX1[n] | Calibration slot base referred to the 1st measuring axis. |
| SD54617 $SNS_MEA_CAL_EDGE_PLUS_DIR_AX1[n] | Calibration slot edge in the positive direction of the 1st measuring axis. |
| SD54618 $SNS_MEA_CAL_EDGE_MINUS_DIR_AX1[n] | Calibration slot edge in the negative direction of the 1st measuring axis. |
| SD54619 $SNS_MEA_CAL_EDGE_BASE_AX2[n] | Calibration slot base referred to the 2nd measuring axis. |
| SD54620 $SNS_MEA_CAL_EDGE_UPPERE_AX2[n] | Upper calibration slot edge referred to the 2nd measuring axis. |

| | |
|---|---|
| SD54621 $SNS_MEA_CAL_EDGE_PLUS_DIR_AX2[n] | Calibration slot edge in the positive direction of the 2nd measuring axis. |
| SD54622 $SNS_MEA_CAL_EDGE_MINUS_DIR_AX2[n] | Calibration slot edge in the negative direction of the 2nd measuring axis. |

**Note**

For a standard lathe with axes X and Z (G18), axis Z is the 1st measuring axis and axis X is the 2nd measuring axis.

### 18.10.4.5 Measuring tools at the turning machines

#### Calibration data of the tool probe referred to the machine coordinate system

If you wish to calibrate the tool probe in the machine coordinate system, then the position of the tool probe in the machine coordinate system must be entered into the following general cycle setting data.

| | |
|---|---|
| SD54625 $SNS_MEA_TP_TRIG_MINUS_DIR_AX1[k] | Trigger point in minus direction of the 1st measuring axis (for G18 Z). |
| SD54626 $SNS_MEA_TP_TRIG_PLUS_DIR_AX1[k] | Trigger point in plus direction of the 1st measuring axis (for G18 Z). |
| SD54627 $SNS_MEA_TP_TRIG_MINUS_DIR_AX2[k] | Trigger point in minus direction of the 2nd measuring axis (for G18 X). |
| SD54628 $SNS_MEA_TP_TRIG_PLUS_DIR_AX2[k] | Trigger point in plus direction of the 2nd measuring axis (for G18 X). |

#### Calibration data of the tool probe referred to the workpiece coordinate system

If you wish to calibrate the tool probe in the workpiece coordinate system, then the position of the tool probe in the workpiece coordinate system must be entered into the following general cycle setting data. In this case, the reference point is the outer diameter or the tool length of the active tool in the spindle.

Index [k] stands for the number of the actual data field (probe number -1).

| | |
|---|---|
| SD54640 $SNS_MEA_TPW_TRIG_MINUS_DIR_AX1[k] | Trigger point in minus direction of the 1st measuring axis (for G18 Z). |
| SD54641 $SNS_MEA_TPW_TRIG_PLUS_DIR_AX1[k] | Trigger point in plus direction of the 1st measuring axis (for G18 Z). |
| SD54642 $SNS_MEA_TPW_TRIG_MINUS_DIR_AX2[k] | Trigger point in minus direction of the 2nd measuring axis (for G18 X). |
| SD54643 $SNS_MEA_TPW_TRIG_PLUS_DIR_AX2[k] | Trigger point in plus direction of the 2nd measuring axis (for G18 X). |

### Tool measurement using the "Orientable tool holder" or "Swivel tool" function

If the general cycle machine data MD52740 $MNS_MEA_FUNCTION_MASK, bit 16 = 1, then the following setting applies:

| MD51618 $MNS_MEA_CM_ROT_AX_POS_TOL | | Tolerance parameter for rotary axis settings |
|---|---|---|
| = 0.5 | Default value | |

The real angular position of rotary axes can deviate from that programmed (exact stop fine window) This deviation depends on the position control properties of the axis. The maximum deviation that can be expected at a specific axis should be entered into the parameter. When the tolerance is exceeded, Alarm 61442 is output - "Tool holder not parallel to the geometry axes".

## 18.10.5 Logging

### 18.10.5.1 Logging the measuring, general

General logging can be performed without special settings:

- In the passive file system of the NC
- On the local drive of the CF card

If you want to use a logical drive for the logging, you require the Execution from External Storage (EES) function.

A detailed description of the EES function is provided in the following documentation:

### References

Function Manual Basic Functions; Mode Group, Channel, Program Operation, Reset Response (K1)
Chapter "EES (Execution from External Storage)"

### General machine data

Logging can be controlled via the following machine data settings:

| MD11420 $MN_LEN_PROTOCOL_FILE | | File size for log files (KB) |
|---|---|---|
| ≥ 20 | recommended value | |

Increase the value of machine data MD11420 if larger log files have to be stored in the file system of the NC.

| MD11422 $MN_PROTOCOL_FILE_MODE | | Setting the behavior of the WRITE command |
|---|---|---|
| Bit 0 | Persistence behavior | |
| = 0 | time-delayed WRITE command (immediately persistent) | |
| = 1 | = fast WRITE command (time-delayed persistent) | |

**Note**

Set MD11422 bit 0 = 1 so that the program runtimes are not lengthened significantly.

### 18.10.5.2 Logging while measuring in the JOG mode

You can set the following setting data in the "Settings for Measurement Log" window for the JOG mode.

#### References

Operating Manual, Turning/MIlling/Grinding; Chapter "Settings for the measurement result log"

#### Cycle channel setting data

| SD55774 $SCS_J_MEA_PROTOCOL_FILE | Name and path for the log file for measuring in JOG |
|---|---|
| //NC:/WKS.DIR/TEMP.WPD/J_MEAPROT.TXT | Default setting |

| SD55740 $SCS_MEA_FUNCTION_MASK | Measuring cycles function screen, workpiece/tool measurement |
|---|---|
| Bit 28 | Select new log or continuos log for measuring standard log in JOG |
| = 0 | New log (default value) |
| = 1 | Continuous, i.e. always attach to the existing log |
| Bit 29 | Select file format for measuring standard log in JOG |
| = 0 | Text format (default value) |
| = 1 | Tabular format |

### 18.10.5.3 Logging while measuring in the AUTOMATIC mode

#### Cycle channel setting data

| SD55730 $SCS_MEA_PROTOCOL_USER_EXT | File extension for user log |
|---|---|
| TXT | Default setting |

| SD55614 $SCS_MEA_RESULT_MRD | Setting of the program control of the measurement result display MRD |
|---|---|
| = 0 | Switch off the measurement result display |
| = 1 | Switch on the measurement result display (default value) |

### Definition file: User GUD block

Data for user logs are generally logged from the S_PROTTXT[10] GUD array (PGUD).

If the data array is not sufficient, you can create a second array with the predefined name S_USERTXT[n] in a separate GUD block (e.g. MGUD or UGUD):

```
DEF NCK STRING[200] S_USERTXT[n]  ; n = number of field elements
```

**Note**

The logging operation checks whether the S_USERTXT array is available. If S_USERTXT is available, the content of the array is logged. If S_USERTXT is not available, the content of the S_PROTTXT array is logged.

## 18.10.5.4 Manufacturer and user cycle CUST_MEAPROT.SPF

### Function

You can make individual adaptations to the log form and content in the CUST_MEAPROT cycle.

### Procedure

1. Open the CUST_MEAPROT.SPF cycle file.
2. Select the program lines in the _MARK1 block.
3. Copy the contents from the _MARK1 block to the _MARK2 block.
4. Adapt the parameters individually in the _MARK2 block.
5. Save the changed cycle file.
   The original default values are retained and can be activated/deactivated at any time (setting of comment characters).

### Parameter

In the CUST_MEAPROT cycle, all the possible parameter changes are described in detail in the _MARK2_1 block.

You can adapt the following parameters:

| Parameter | Default value | Input area |
|---|---|---|
| Characters per line | 66 | > 66 possible (1 |
| Characters per column 2 to 4 | 14 | Min. 12, max. 14 |
| Workpiece counter | "" | Identifier |
| Column space | " " | Blank string (≥ 2 * blanks) |
| Identifier for tolerance specification | "#" | 1 character |

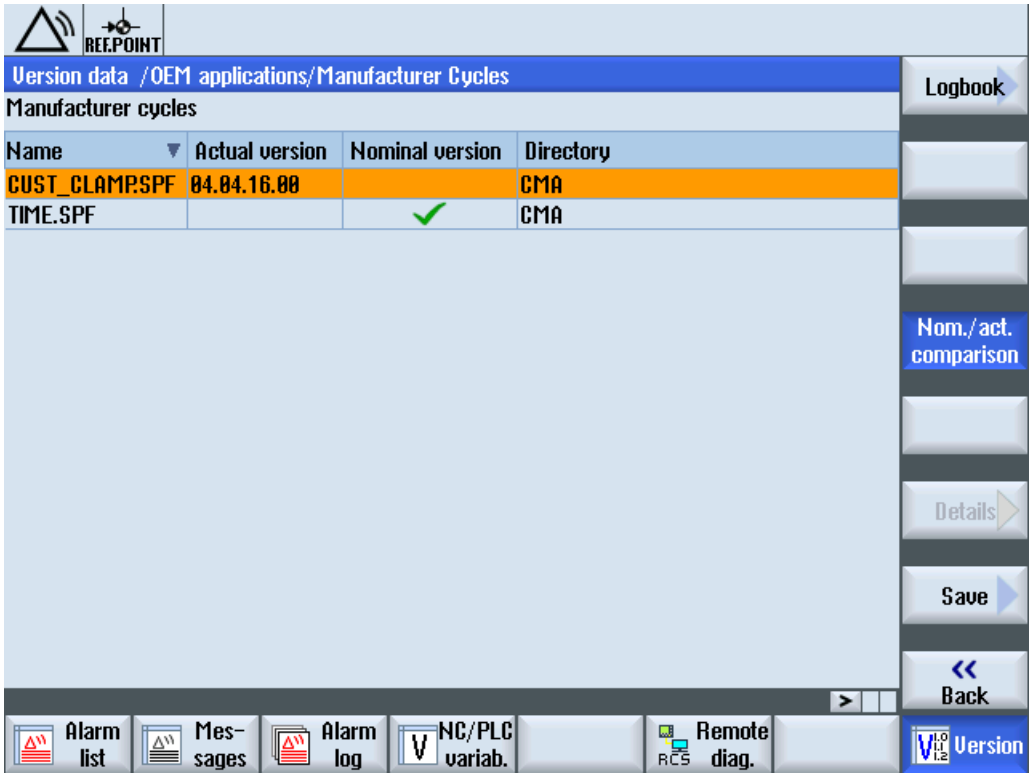
| Parameter | Default value | Input area |
|---|---|---|
| Identifier for tolerance violation | "!" | 1 character |
| Log header lines 5 to 7 | "" | Any text |

1) 66 characters per line correspond to the screen width in the default setting. A log line is therefore displayed completely on the screen. If logging is mainly performed in the file system of the NC, this setting should be retained because of the better readability. If you perform logging mainly in external files, you can increase the line width accordingly. In this case, the width of column 1 is increased automatically (with the same width for columns 2, 3 and 4 and the same column space).

# 18.11 Compare cycles version

## 18.11.1 Display cycles version

If cycles have version details, you can display them in the version screen.




### Precondition

The version details are included in the cycles files in the following form:

`;VERSION:` <Version> `;DATE:` <YYYY-MM-DD>

**Example:**

```
;VERSION: 05.05.05.00 ;DATE: 2012-11-30
```

## Procedure




1. Select the "Diagnostics" operating area.




2. Press the "Version" softkey.

   It takes some time to call the version display. While the version data is being determined a progress message box and the appropriate text are displayed in the dialog line.




3. Select the "OEM applications /manufacturer cycles" areas and press the "Details" softkey.

   The "Manufacturer Cycles" window opens.

   The "Planned-actual comparison" softkey is selected by default.

## See also

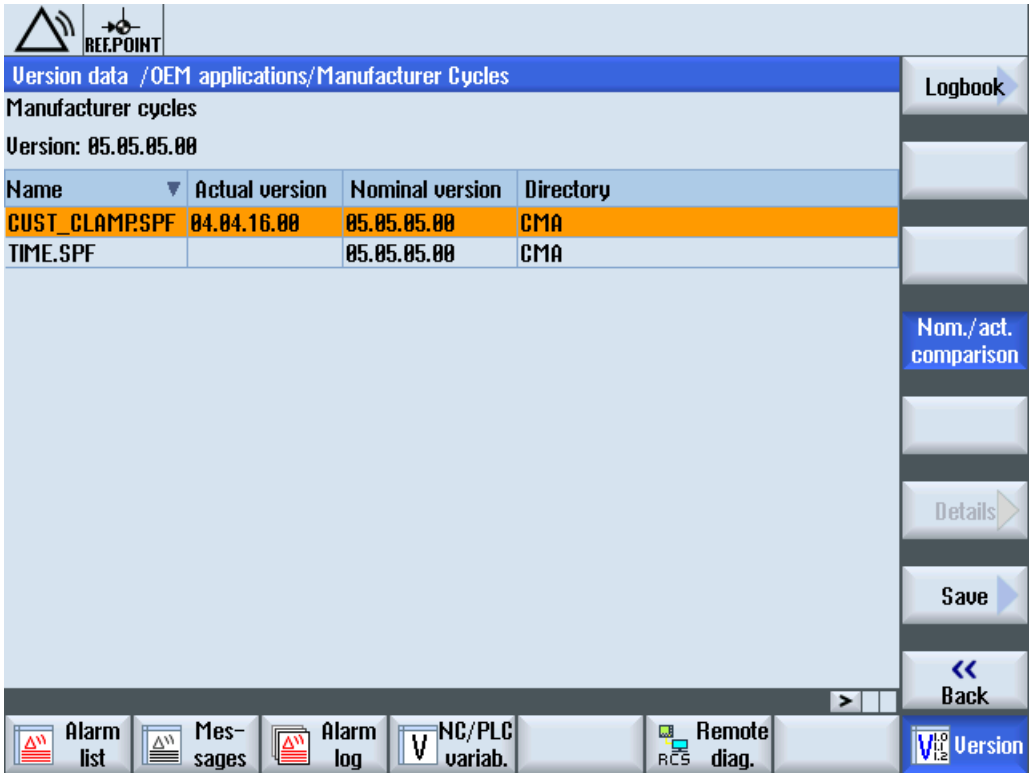
Save information (Page 279)

Logbook (Page 282)

## 18.11.2 Specify cycles version

### General

You have the possibility to compare a required cycles version with the existing cycles version in the version view.




### File overview

The following files are required:

| Name | Path | Meaning |
|---|---|---|
| versions.xml | /siemens/sinumerik/hmi/data/version/oem/cma | Version file |
| versions.xml | /oem/ | Version path file |

### Adapting the version file

The cycles version specification requires a manual adaptation of the version file.

1. Copy the "versions.xml" sample file from the following directory:
   /**siemens**/sinumerik/hmi/data/version/oem/cma
2. Save the sample file in its own directory under /oem.
   The standard directory for this purpose is /oem/sinumerik/hmi/data/version/oem/cma.
3. Open the file.

4. Enter the planned version for the cycles files in the <info> tag with "defaultFileVersion" and the details for the complete version in the <Version> tag.
5. Close the file to save the changes.

**Example:**

```
<info defaultFileType=" *.spf *.cpf" defaultFileVersion="05.05.00.00"
linkname="CMA" linkpath="//NC/CMA.DIR" EffDirOrder="CUS CMA">
<Name>Manufacturer cycles</Name>
<Version>5.5.0.0</Version>
<Link>
<Name>CUS</Name>
<Path>//NC/CUS.DIR</Path>
</Link>
</info>
```

## Adapting the version path file

If the version file is saved in the standard directory, and a current oem\versions.xml is available, no changes are required. In all other cases, the path data can be manually changed:

1. Open the versions.xml file under /oem
2. Replace the standard path in the <Path> tag for the "Manufacturers Cycles" with your own path for the version file.
3. Close the file to save the changes.

**Standard:**

```
<Component>
<Name>Manufacturer Cycles</Name>
<Path>/siemens/sinumerik/hmi/data/version/oem/cma</Path>
</Component>
```

**Example:**

```
<Component>
<Name>Manufacturer Cycles</Name>
<Path>/oem/sinumerik/hmi/data/version/oem/cma</Path>
</Component>
```

# Additional language 19

## 19.1 Installing additional languages

### Additional languages

In addition to the already existing standard languages, additional user interface languages are available on a separate DVD.

The integrated online help is either displayed in the relevant language of the country or in English.

Refer to the following chapter for the languages that are available for the user interface: Supported languages (Page 502)

### Precondition



**Software option**

You require the following software option in order to use this function:

"Additional languages"

- After the installation, approx. 256 MB memory must be available on the CompactFlash Card.
- Language extension software "hmi_sl_language_xxx_02.06.00.00.00y.arc".
  xxx = language code, y = internal version ID
  Is supplied on the language extension DVD.

### Installing a language

Install the language the same as for a commissioning archive (ARC). For the procedure, please refer to the following chapter: Reading-in a start-up archive (Page 203)

**Note**

Language extensions software of the previous version is compatible with the current operating software.

## 19.2 Uninstalling languages

If you no longer require a language interface, then you can uninstall this.

**Note**

Standard languages and the currently active language for the operating software cannot be uninstalled.

### Procedure




1. Select the "Startup" operating area.




2. Press the "Change language" softkey.
   The "Language selection" window opens. The language most recently set is highlighted.
3. Switch the cursor to the required language.




4. Press the "Delete language" softkey.
   The "Delete language" window opens.




5. Press the "OK" softkey.
   The operating software uninstalls the selected language.

## 19.3 Supported languages

Table 19-1 Supported languages

| Language | Language code | Standard languages |
|---|---|---|
| Chinese (simplified) | chs | X |
| Chinese (traditional) | cht | |
| Danish | dan | |
| German | deu | X |
| English | eng | X |
| Finnish | fin | |
| French | fra | X |
| Indonesian | ind | |
| Italian | ita | X |
| Japanese | jpn | |
| Korean | kor | |
| Malaysian | msl | |
| Dutch | nld | |
| Polish | plk | |

| Language | Language code | Standard languages |
|---|---|---|
| Portuguese | ptb | |
| Romanian | rom | |
| Russian | rus | |
| Swedish | sve | |
| Slovenian | slv | |
| Slovakian | sky | |
| Spanish | esp | X |
| Thai | tha | |
| Czech | csy | |
| Turkish | trk | |
| Hungarian | hun | |
| Vietnamese | vit | |

# SINUMERIK Operate on PCU/IPC 20

## 20.1 Link OEMFrame application

### Integrating an OEM application

The following files should be adapted and created in order to integrate an OEMFrame application into the operating software:

- Configuration file "systemconfiguration.ini"
  The operating software is started and controlled from the system manager; the system manager also controls the OEMFrame applications. The system manager is configured using the "systemconfiguration.ini" configuration file.
- Configuring the start softkey
  In order to start an OEMFrame application from the operating software, configure a softkey on the expansion bar of the operating area.
- Configuration file "slamconfig.ini"
  In order to configure the softkey position with text and/or symbol for the OEMFrame application, generate the "slamconfig.ini" file.
- Save the language-dependent text for a softkey in the file mytext_<lng>.ts.

### "systemconfiguration.ini" file

In order to integrate an OEMFrame application into the operating software, copy the "systemconfiguration.ini" configuration file and place it in one of the two directories:

<Installation path>/**user**/sinumerik/hmi/cfg

<Installation path>/**oem**/sinumerik/hmi/cfg

All of the processes to be managed by the system manager, as well as the applications that are to be integrated as OEMFrame applications are in the **[processes]** section.

| Value | Meaning |
|---|---|
| `process` | Symbolic name of the OEMFrame application. This is required to configure the operating areas. |
| `cmdline` | Command line which is transferred to the "oemframe.exe" process when starting. |
| `oemframe` | For OEMFrame applications, always set to "true". |
| `windowname` | Window name of the OEMFrame application - determined with "findwindow.exe". |
| `classname` | Class name of the OEMFrame application - determined with "findwindow.exe". |
| `deferred` | `true`: OEMFrame application is not started when SINUMERIK Operate powers up, but only when selected for the first time. |

| Value | Meaning |
|---|---|
| `startupTime` | The linked process starts as follows:<br>`immediately`: immediately (default)<br>`afterServices`: after all services have started<br>`afterGuis`: after all GUI components have started<br>If SINUMERIK Operate is shut down, the process goes in reverse:<br>`immediately`: immediately (default)<br>`afterServices`: after all services have shut down<br>`afterGuis`: after all GUI components have shut down |
| `gimmekeys` | Release mask for system configuration keys that are to be handled by the OEM-Frame application. The parameterization is performed in the form of a bit mask. |
| `disablekeys` | Parameterization for the keyboard filter behavior. The parameterization is performed in the form of a bit mask. |
| `menuselectkey` | The value is used for changing the key that calls the operating area menu (default F10). The value is a logically ORed combination of the modifications (Microsoft definition) Key_Shift, Key_Alt, Key_Ctrl, and the virtual key code. |
| `timeout` | Maximum duration for the search for the OEMFrame application in milliseconds. If the OEMFrame application was not found within this time, it is not managed by the System Manager.<br>Default setting in the "systemconfiguration.ini" file:<br>`[miscellaneous]`<br>`startTimeoutDefault` |
| `shutdowntime` | Maximum duration for downloading the OEMFrame application in milliseconds. If the OEMFrame application was not found after this period of time, the process is canceled.<br>Default setting in the "systemconfiguration.ini" file:<br>`[miscellaneous]`<br>`shutdownTimeoutDefault` |

## Bit mask "gimmekeys"

The bit mask for an OEMFrame application is set to the binary value 0xF by default. All of the key combinations from F1 to F8 are fed to the OEMFrame application. The OEMFrame application itself can handle a specific key/key combination by setting additional bits. Otherwise, the system configuration will evaluate the key/key combination and it will not even reach the OEMFrame application.

You can parameterize the bit mask "gimmekeys" as follows:

| Bit | Keys | Meaning |
|---|---|---|
| 0 | F1 - F8 | Horizontal softkeys (upper bar, HU) |
| 1 | Shift+F1 - Shift+F8 | Vertical softkeys (right bar, VR) |
| 2 | Ctrl+F1 - Ctrl+F8 | Horizontal softkeys (lower bar, HL) |
| 3 | Shift+Ctrl+F1 - Shift+Ctrl+F8 | Vertical softkeys (left bar, VL) |
| 4 | F9 | Recall |
| 5 | Shift+F9 | ETC switchover |
| 6 | F10 | Operating area menu |
| 7 | Shift+F10 | M key |

| Bit | Keys | Meaning |
|---|---|---|
| 8 | F11 | Channel switchover key |
| 9 | Shift+F11 | M key (hard key) |
| 10 | F12 | Info/help |
| 11 | Shift+F12 | Custom key (hard key) |
| 12 | ESC | Alarm cancel |
| 13 | HOME | Window switchover key |
| 14 | END | PROGRAM (hard key) |
| 15 | PAGE UP | ALARM (hard key) |
| 16 | PAGE DOWN | TOOL OFFSET (hard key) |
| 17 | HOME (NUMPAD) | PROGRAM MANAGER (hard key) |
| 18 | F13 - F20 | Extended, horizontal softkeys (upper bar, HU) |
| 19 | Shift+F13 - Shift+F20 | Extended, vertical softkeys (right bar, VR) and right direct keys HT8 |
| 20 | Ctrl+F13 - Ctrl+F20 | Extended, horizontal softkeys (lower bar, HL) |
| 21 | Shift+Ctrl+F13 - Shift+Ctrl+F20 | Extended, vertical softkeys (left bar, VL) and left direct keys HT8 |

## Bit mask "disablekeys"

The bit mask for an OEMFrame application is set to the binary value 0x3FFFF by default. Thus, all of the keyboard sequences are filtered out and not forwarded to the OEMFrame application. If a bit is set to 0, the keyboard filter for the corresponding sequence is deactivated and the OEMFrame application is able to receive the keyboard sequence.

If, for example, an OEMFrame application is to receive all of the softkeys of left and bottom softkey bar, set the "disablekeys" bit mask to the binary value 0x300FF.

You can parameterize the bit mask "disablekeys" as follows:

| Bit | Keys | Meaning |
|---|---|---|
| 0 - 7 | Reserved | |
| 8 | (Shift)+Ctrl+F1 | Lower and left-hand softkey bar (HL, VL) |
| 9 | (Shift)+Ctrl+F2 | Lower and left-hand softkey bar (HL, VL) |
| 10 | (Shift)+Ctrl+F3 | Lower and left-hand softkey bar (HL, VL) |
| 11 | (Shift)+Ctrl+F4 | Lower and left-hand softkey bar (HL, VL) |
| 12 | (Shift)+Ctrl+F5 | Lower and left-hand softkey bar (HL, VL) |
| 13 | (Shift)+Ctrl+F6 | Lower and left-hand softkey bar (HL, VL) |
| 14 | (Shift)+Ctrl+F7 | Lower and left-hand softkey bar (HL, VL) |
| 15 | (Shift)+Ctrl+F8 | Lower and left-hand softkey bar (HL, VL) |
| 16 | Reserved | |
| 17 | Reserved | |

## Bit mask "disablekeyshigh"

It may be necessary to map key sequences because the operating system already responds to Ctrl-F4 and Ctrl-F6 in certain situations.

You can parameterize the bit mask "disablekeyshigh" as follows:

| Bit | Meaning |
|---|---|
| 0 - 28 | Reserved |
| 29 | Key sequences Ctrl-F1 to Ctrl-F8 can be mapped onto key sequences Ctrl-F13 to Ctrl-F20. |
| 30 - 31 | Reserved |

**Note**

The "gimmekeys", "disablekeys" and "disablekeyshigh" bit masks can be specified both with a decimal code (e.g. 31) and a hexadecimal code (e.g. 0x1F).

## Examples

**Note**

**Write error**

Avoid write errors. Determine the entries for the sections "processes" and "areas" only with the FindWindow program.

### "notepad.exe" and "calc.exe"

In the following example, the two Windows applications "notepad.exe" and "calc.exe" are configured as OEMFrame applications:

```
[processes]

PROC500=process:=notepadOEM, cmdline:="C:\\WINDOWS\\system32\
\notepad.exe", oemframe:=true, deferred:=true,
windowname:="Untitled - Notepad", classname:="Notepad"


PROC501=process:=calcOEM, cmdline:="C:\\WINDOWS\\system32\
\calc.exe", oemframe:=true, deferred:=true,
windowname:="Calculator", classname:="SciCalc"


[areas]

AREA500=name:=AreaNote, process:=notepadOEM

AREA501=name:=AreaCalc, process:=calcOEM
```

### "keycatcher.exe"

The Windows application "keycatcher.exe" is integrated in the following example. In this case, all four softkey bars and the Recall key will reach the Windows application. The keyboard filter for the lower and the left-hand softkey bar is deactivated:

```
[processes]
```

```
PROC500= process:=keycatcherOEM, cmdline:="keycatcher.exe",
oemframe:=true, deferred:=true, windowname:="keycatcher",
classname:="QWidget", gimmekeys:=0x1F, disablekeys:=0x300FF


[areas]

AREA500=name:=AreaKeyCatcher, process:= keycatcherOEM
```

The Windows application "keycatcher.exe" is integrated in the following example. In this case, all four softkey bars and the F10 key will reach the Windows application. Press Ctrl+F12 to show the operating area menu when a Windows application is displaying (the system configuration no longer evaluates F10):

```
[processes]

PROC500= process:=keycatcherOEM, cmdline:="keycatcher.exe",
oemframe:=true, deferred:=true, windowname:="keycatcher",
classname:="QWidget", gimmekeys:=0x4F, disablekeys:=0x300FF,
menuselectkey:=Key_Control|0x7B


[areas]

AREA500=name:=AreaKeyCatcher, process:= keycatcherOEM
```

The Windows application "keycatcher.exe" is integrated in the following example. In this case, all four softkey bars are fed to the Windows application. Key sequences Ctrl-F1 to Ctrl-F8 can be mapped onto key sequences Ctrl-F13 to Ctrl-F20:

```
[processes]

PROC500= process:=keycatcherOEM, cmdline:="keycatcher.exe",
oemframe:=true, deferred:=true, windowname:="keycatcher",
classname:="QWidget", gimmekeys:=0xF, disablekeys:=0x300FF


[areas]

AREA500=name:=AreaKeyCatcher, process:= keycatcherOEM
```

## The [areas] section

The SINUMERIK Operate operating areas are configured in this section.

| Value | Meaning |
|---|---|
| name | Symbolic name for the operating area |
| process | Name of the OEMFrame application according to section [processes] |
| panel | Name of the panel (header) to be used<br>Only "SlHdStdHeaderPanel" is currently available for OEMFrame applications. |
| plcid | ID to identify the operating area using the SINUMERIK Operate monitor<br>Only values in the range of 150 to 199 are permitted. |

| NOTICE |
| --- |
| **Basic components are overwritten**<br>If you use numbers less than 500, then it is possible that Siemens basis components will be overwritten. In the "processes" and "areas" sections, the number range 500 - 999 is permissible. |

**Example**

```
[areas]
AREA600= name:=AreaOEM, process:=notepadOEM
AREA601= name:=AreaCalc, process:=calcOEM, panel:=SlHdStdHeaderPanel
```

**Note**

Only OEMFrame applications are supported that use programming interfaces of SINUMERIK Operate.

## The [miscellaneous] section

You can make various settings in this section. Generally, only the start operating area is changed.

| Key | Value |
| --- | --- |
| `startuparea` | Name of the start operating area |

**Example**

```
[miscellaneous]
startuparea = AreaOEM
```

## Configuring the operating area menu

The operating area menu is intended for switching over the operating areas configured in the "systemconfiguration.ini" configuration file. A softkey for selecting the appropriate operating area is provided on the horizontal softkey bar for each operating area configured.

The operating area displays the names of the operating areas from the "systemconfiguration.ini" configuration file as the text on the operating area softkeys. The system automatically searches for a free softkey on the horizontal softkey bar for each operating area.

## Configuring additional settings

In order to configure the following settings you require the "slamconfig.ini" configuration file:

- Assigning a softkey position to a specific operating area.
- Creating a language-dependent text for the softkey.
- Displaying a symbol for the operating area on the softkey.

## Creating the "slamconfig.ini" configuration file

Copy the "slamconfig.ini" and place the file in the same directory in which the "systemconfiguration.ini" file is located:

<Installation path>/**user**/sinumerik/hmi/cfg

<Installation path>/**oem**/sinumerik/hmi/cfg

## "slamconfig.ini" file

In the "slamconfig.ini" configuration file, for every operating area, you can create a section that was configured in the "systemconfiguration.ini" file. The section must bear the configured name of the appropriate operating area, e.g. AreaOEM.

| Value | Meaning |
|---|---|
| `TextId` | Text ID for a foreign-language text which will be displayed as the softkey label. |
| `TextContext` | Context of the foreign-language text. |
| `TextFile` | Name of the text file which includes the context and the foreign-language text. |
| `Graphic` | Name of an image file which will be used as an icon for the softkey. |
| `SoftkeyPosition` | Fixed softkey position of the area softkey. In this case, softkey positions 1 to 8 are located on the 1st horizontal bar and softkey positions 9 to 16 on the 2nd horizontal bar, etc. |
| `AccessLevel` | Access level from which the softkey will be displayed. If this value is not specified, the access level 7 (keyswitch position 0) is set. |

## Example

The softkey for the "AreaOEM" operating area with the following properties is configured in the following example:

- The softkey displays the text which has been stored in the "mytext_<lng>.ts" text file under the context "mycontext" with the "MY_AREA" TextID.
- The "mypicture.png" icon is displayed on the softkey.
- The softkey is located at position 7 in the operating area menu.
- The softkey with access level 5 (keyswitch position 2) is displayed.

```
[AreaOEM]
; Text-ID of a language dependent text
TextId = MY_AREA
; File name of the text file which contains the Text-ID
```

```
TextFile = mytext
; Context in the text file to which the Text-ID is assigned to
TextContext = mycontext
; File name of an icon shown on the area softkey
Picture = mypicture.png
; Position of the area softkey on area menu,
; If no position is specified, an empty position is searched
SoftkeyPosition = 7
; Access level of the area softkey
AccessLevel = 5
```

**Note**

Operating area position 7 is reserved for OEM users.

## Labeling text for the softkey

Storage path:

<Installation path>/**user**/sinumerik/hmi/lng

<Installation path>/**oem**/sinumerik/hmi/lng

The XML identifiers have the following meanings:

| Attribute | Description |
|---|---|
| context | Context within the text file.<br>Each file must have at least one context. |
| name | Name of the context. |
| message | Text translation.<br>There must be at least one message per context. |
| source | Text identifier. |
| translation | Translated text. |
| remark | Text comment (optional). |
| chars | Maximum possible length of the text in characters. If nothing is specified, the text can have any length (optional). |
| lines | Maximum number of lines available for display. If nothing is specified, the number of lines is unlimited (optional). |

Structure of the language-dependent ts file that contains the labeling text for the softkey:

**mytext_<lng>.ts**

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!DOCTYPE TS>
```

**mytext_<lng>.ts**

```
<TS>
  <context>
  <name>mycontext</name>
    <message>
      <source>MY_AREA</source>
        <translation>Text, which is displayed on the softkey</translation>
        <remark>comment (optional)</remark>
        <chars>20</chars>
        <lines>2</lines>
    </message>
  </context>
</TS>
```

<lng> stands for the language code.

## 20.2 Parameterizing the OEMFrame application

### File "oemframe.ini"

Using the "oemframe.ini" file, it is possible to further parameterize OEMFrame applications. Create the file in the following directory:

<Installation path>/compat/oem

Create a separate section with the required parameters for each OEMFrame application. Name the section according to the corresponding program file without a filename extension. Place the name in square brackets.

#### Example

```
[notepad]
```

### Parameter overview

The following parameters can be used for OEM applications:

| Parameter | Meaning | Default value |
|---|---|---|
| WindowStyle_On | Defines properties to be assigned to the window | 0 |
| WindowStyte_Off | Defines properties that the window must not have | 0 |
| x | Horizontal starting coordinate of the OEMFrame application (unit: pixels) | 0 |
| y | Vertical starting coordinate of the OEMFrame application (unit: pixels) | 0 |

| Parameter | Meaning | Default value |
|---|---|---|
| Width | Width of the OEMFrame application (unit: pixels) | Desktop width |
| Height | Height of the OEMFrame application (unit: pixels) | Desktop height |
| nDelayInitComplete | Delays the feedback to the System Manager | 0 |
| fSearchOnlyForTaskWindow | States whether the window specified in the "systemconfiguration.ini" file belongs to the task specified there | 1 |
| fRestoreTaskWindow | Defines the behavior when exiting an application that was started from the OEMFrame application | 0 |
| fKeepPlacement | Deactivates the size adaptation | 0 |
| fForceTaskFocus<br>fSearchForPopUps | Define which window of the OEMFrame application will be displayed when starting. | 0<br>1 |
| nInitShowMode | State in which the window of the OEMFrame application is displayed when the application is started. | SW_SHOWMINNOACTIVE |
| nShowMode | State in which the window of the OEMFrame application is displayed when it is shown. | SW_SHOWNORMAL |
| nUnShowMode | State into which the window of the OEMFrame application is put when it is hidden. | SW_SHOWMINNOACTIVE |
| fWinForms | Must be set if the application is a "Windows Forms Application" | 0 |
| nSwitchToTaskAfterTermination | Controls the behavior when the OEMFrame application is exited | -1 |
| fFindWindowWithWildcards | Used for wildcards (?, *) for the attribute Windowname | 0 |

## WindowStyle_On/WindowStyle_Off

The appearance of a Windows application is defined with the aid of the Windows API function SetWindowLong. When the SetWindowLong function is called, the appearance of the application is controlled using an 8-byte word. 2 bytes of this word can be changed using the WindowStyle_On and WindowStyle_Off parameters.

The WindowStyle_On parameter defines properties to be assigned to the window. The WindowStyle_Off defines properties that the window must not have.

The following control options are possible with WindowsStyle parameters (binary characteristic values):

| 0000 | 0000 | xxxx | xxxx | 0000 | 0000 | 0000 | 0000 |
|---|---|---|---|---|---|---|---|
| | | 1010 | | Header | | | |
| | | 1000 | | Border | | | |
| | | 0100 | | Type of window frame on a dialog box | | | |
| | | 0010 | | Vertical scroll bar | | | |
| | | 0001 | | Horizontal scroll bar | | | |
| | | | 1000 | System menu | | | |
| | | | 0100 | Frame (Thickframe) | | | |
| | | | 0010 | Minimize box | | | |
| | | | 0001 | Maximize box | | | |

Assign binary characteristic values to the WindowStyle parameters as decimal numbers. It is possible to do the conversion using the calculator in Windows, for example.

#### Example conversion

The properties of the system menu as well as the horizontal and vertical scroll bar should be defined. According to the table, these are:

```
0000 0000 0011 1000 0000 0000 0000 0000 binary or
0038 0000 hexadecimal
```

1. Select Hex on the calculator.
2. Enter the digit string 00380000 (leading zeroes can be omitted).
3. Select Dec.
   Result: 3670016
4. Assign the result to the parameter as a characteristic value.

#### Application examples

For the Windows application Notepad, the system menu as well as the horizontal and vertical scroll bar should be activated:

```
[notepad]
WindowStyle_On = 3670016
```

No minimize box and no maximize box should be displayed for the Windows application Notepad:

```
[notepad]
WindowStyle_Off = 196608
```

### x/y

The parameters `x` and `y` define the starting coordinates of the window of the Windows application to be linked, measured from an origin in the upper left corner of the screen. `x` is the horizontal coordinate, `y` is the vertical coordinate pointing downward. The unit of measurement is pixels.

The available working area depends on the screen layout being used.

### Width

This parameter defines the width of the window for the Windows application, measured in pixels from the window origin according to parameter `x`.

### Height

This parameter defines the height of the window for the Windows application, measured in pixels from the window origin according to parameter `y`.

### nDelayInitComplete

As soon as the window of a Windows application has been initiated, the information is sent to the System Manager. The Windows application can then be selected using the System Manager. This information can be delayed using the `nDelayInitComplete` parameter. The unit is specified in milliseconds.

A delay is necessary, for example, if the Windows application must carry out other actions while generating its window. If the window is activated too soon by the system manager, display errors will occur in the Window.

#### Example

After creating its window, the Windows application "app.exe" reads additional status data from a database. The window of the Windows application may only be displayed after all status data has been read. This read operation should take approximately one second. The following parameter assignment is possible:

```
[app]
nDelayInitComplete = 2000
```

### fSearchOnlyForTaskWindow

This parameter specifies whether the window specified in the file "systemconfiguration.ini" using `ClassName`/`WindowName` belongs to the task that is also specified there.

The following values can be used:

| `fSearchOnlyForTaskWindow` | |
|---|---|
| `= 0` | The window does not belong to the specified task. |
| `= 1` | The window belongs to the specified task. |

When searching, not only are the windows of the task configured in "systemconfiguration.ini" taken into account, but all of the windows that exist in the system at the time of the search.

#### Example

The Windows application comprises more than one process, e.g. a "startup.exe" and a "user.exe". In the "systemconfiguration.ini" file, only "startup.exe" is entered, from which "user.exe" is started. The application window belongs to "user.exe" and is therefore not found when searching is restricted to the windows of "startup.exe".

The following parameter assignment is possible:

```
[startup]
fSearchOnlyForTaskWindow = 0
```

## fRestoreTaskWindow

This parameter defines the behavior when exiting a Windows application that was started from the OEMFrame application as second task level.

When the OEMFrame application is deselected, the last window that was active (`ForegroundWindow`) is saved by default. When reselecting the OEMFrame application, this window is reactivated.

If another application was started from the OEMFrame application, then the active window refers to the new Windows application.

**Note**

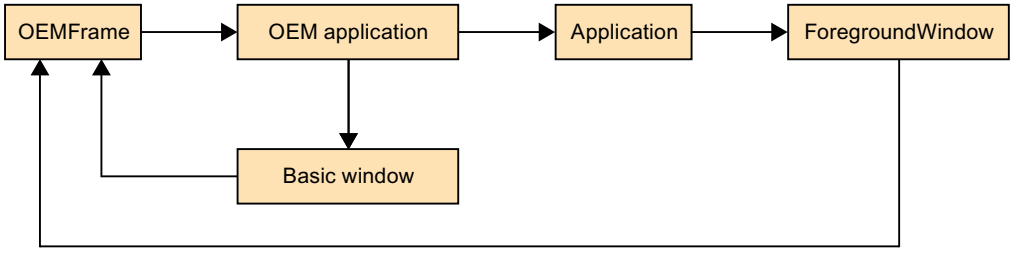
**Incorrect displays**

In many cases, the proxy application "oemframe.exe" cannot detect when the new Windows application ends. Display errors can occur in the second task level. Activate a basic window for the OEMFrame application.

The behavior can be influenced via the following values:

| `fRestoreTaskWindow` | |
|---|---|
| `= 0` | When selecting the OEMFrame application or starting a Windows application from the OEMFrame application, the `ForegroundWindow` is activated. When ending the second task level, the OEMFrame application is displayed. |
| `= 1` | When selecting the OEMFrame application or starting a Windows application from the OEMFrame application, a basic window is activated before the `ForegroundWindow` is activated. When ending the second task level, the basic window of the OEMFrame application is always displayed. |

### Schematic




## fForceTaskFocus/fSearchForPopUps

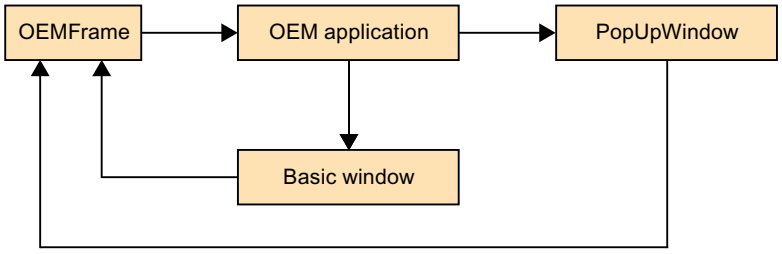
These two parameters define which window the OEMFrame application will activate after it has been deselected and reselected again.

On switching from one operating area to another, the last active window `ForegroundWindow` of the OEMFrame application is saved. When reselecting the application, the `ForegroundWindow` window is then reactivated.

The behavior can be changed via the following values:

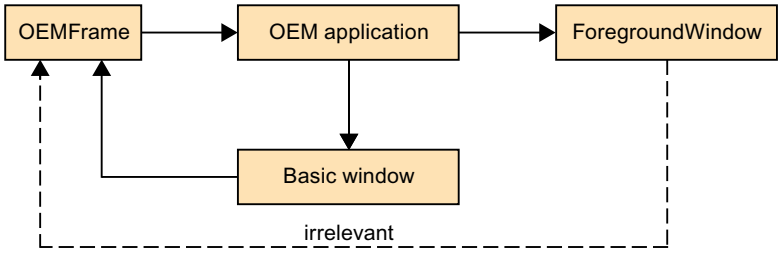
| `fForceTaskFocus` and `fSearchForPopUps` | |
|---|---|
| `= 1` | When the OEMFrame application is deselected, an active popup window is searched for rather than the `ForegroundWindow` window.<br>Behavior of the search:<br>• If a popup window was found, it is displayed when the OEMFrame application is reselected.<br>• If no popup window exists or was found, the basic window of the application is displayed when the application is reselected. |

**Schema fForceTaskFocus/fSearchForPopUps with value 1**




With the following value change, only the basic window of the OEMFrame application is taken into consideration when an application is selected or deselected. The basic window is specified in `ClassName`/`WindowName` in the "systemconfiguration.ini" file:

**Schema fForceTaskFocus with value 1 and fSearchForPopUps with value 0**




## fKeepPlacement

This parameter deactivates the resizing (zooming) for the basic window of the OEMFrame application. By default, the application is zoomed to the screen size before it is displayed. In the case of applications that do not allow their windows to be zoomed, the resizing can result in display problems. In such cases, the zoom function must be deactivated.

### Example

An application "fixres.exe" should be displayed in its programmed window size. The following parameter assignment is required:

```
[fixres]
fKeepPlacement = 1
```

| fSearchForPopUps | |
|---|---|
| = 0 | No active pop-up window is searched for. |

## nInitShowMode/nShowMode/nUnShowMode

The three parameters define how the application window will be displayed when the application is started, hidden, and shown.

The parameter nInitShowMode starts the application. The nShowMode parameter refers to showing the application (area is activated). The parameter nUnShowMode hides the application.

The following value range exists for the parameters nShowMode and nUnShowMode:

| nShowMode and nUnShowMode | |
|---|---|
| = 0 | The application window is hidden (SW_HIDE). |
| = 1 | The application window is displayed in its original form (position, size) and has the input focus (SW_SHOWNORMAL, SW_NORMAL). |
| = 2 | The application window is minimized and has the input focus (SW_SHOWMINIMIZED). |
| = 3 | The application window is maximized (SW_SHOWMAXIMIZED). |
| = 4 | The application window is displayed without having the input focus (SW_SHOWNOACTIVATE). |
| = 5 | The application window is displayed and has the input focus (SW_SHOW). |
| = 6 | The application window is minimized and loses the input focus (SW_MINIMIZE). |
| = 7 | The application window is minimized without having the input focus (SW_SHOWMINNOACTIVE). |
| = 8 | The application window is displayed without having the input focus (SW_SHOWNA). |
| = 9 | The application window is displayed in its original form (position, size) (SW_RESTORE). |
| = 10 | The application window is displayed in the same way as when the application was started (SW_SHOWDEFAULT). |

**Note**

**Display problems**

The default settings function for most of the applications.

In exceptional cases, display problems can occur in applications with Borland-Delphi development, i.e. shifted windows, etc.

Use the following parameters in this case:

```
nUnShowMode = 0
```

```
fKeepPlacement = 1
```

or

```
nInitShowMode = 1
```

## fWinForms

Windows Forms parameters provide control elements to applications which also use Windows default applications, e.g. dialog boxes, menus and buttons.

If the OEMFrame application is a Windows Forms application, then the following parameterization is required:

```
[<app-name>]
```

```
fWinForms = 1
```

**Note**

**Deactivated size adjustment**

If the parameter `fWinForms` is not set, the OEMFrame application does not open to the maximum size. The specified size adjustments (`x`, `y`, `Width` and `Height`) are deactivated.

## nSwitchToTaskAfterTermination

This parameter controls the behavior when the OEMFrame application is exited. Normally in this situation, the system manager displays the operating area switchover, which can be used to switch over to another area.

The parameter `nSwitchToTaskAfterTermination` can be used to initiate an automatic switch to the previously active area:

| `nSwitchToTaskAfterTermination` | |
|---|---|
| `= -1` | Display of the operating area switchover (default) |
| `= -2` | Switchover to the previously active area |

**Example**

When the "closeapp.exe" Windows application is exited, a switch should be made to the previous area. The following parameter assignment is required:

```
[closeapp]
nSwitchToTaskAfterTermination = -2
```

### fFindWindowWithWildcards

With this parameter, a wildcard search can be activated for the attribute `Windowname`. In this way it is possible to identify applications that do not have the same attribute `window name` at every start, e.g. access level, time-of-day under `window name`, etc.:

| fFindWindowWithWildcards | | |
|---|---|---|
| = 0 | Wildcard search is switched off | |
| = 1 | Wildcard search is switched on<br>Wildcard: | |
| | ? | One arbitrary character |
| | * | Any number of characters |

#### Example

An application is to be integrated that contains the start time in the attribute `window name`:

**oemframe.ini**

```
[MyQtTest]
fFindWindowWithWildcards = 1
```

**systemconfiguration.ini**

```
[processes]
PROC500=process:=ProcessOEM, cmdline:="C:\\Program Files(x86)\\Siemens\\
MotionControl\\oem\\sinumerik\\hmi\\appl\\MyQtTest.exe",oemframe:=true,
windowname:="MyQtTest - ??:??:??", classname:="QWidget",deferred:=true

[areas]
AREA500=name:=Test, process:=ProcessOEM
```

## 20.3 FindWindow program application

Via the FindWindow program, you can determine all the parameters in the "systemconfiguration.ini" file that are relevant for integrating an OEMFrame application into SINUMERIK Operate.

### Requirement

To use the FindWindow program, you require access level 2 (service).

### Installation

FindWindow can be used on any PCU without being installed.

FindWindow (findwindow.exe) is located in the following directory:
C:\Program Files (x86)\Siemens\MotionControl\Common\FindWindow\

### Application

1. Start the PCU.
2. Install the application to be integrated using OEMFrame.
3. Open the application.
4. Open the FindWindow program.
5. Left-click to drag the red cross to the title bar of the application in the FindWindow program.
6. Check the application parameters "FileName", "ClassName", etc.
7. Select the parameters "Gimmekey", "Disablekeys", etc. via the checkboxes as needed.

   If the configuration file "systemconfiguration.ini" exists and you want to expand it:
8. Copy the entries to the clipboard by clicking on "Copy to Clipboard".
9. Open the existing configuration file and insert the parameters from the clipboard. To do this, right-click on the position at which the parameters are to be inserted and then click on "Insert".
10. Save the modified configuration file.

    - OR -

    If the configuration file "systemconfiguration.ini" does not exist and you want to create it:
8. Click on the "Create systemconfiguration.ini" button.

   A new configuration file is simultaneously created in the directory for FindWindow. Any existing configuration file will be overwritten.
9. Insert a copy of the new configuration file into the provided directory.

**Note**

If you include a 64-bit application, no parameter data is displayed under "FileName". In this case, an information dialog opens, which allows you to manually select the application.

**Note**

Only use the 8.3 Notation for filenames if problems should occur without this parameterization.

## 20.4 Activating/deactivating window mode

By default, SINUMERIK Operate is displayed without window mode on PCU/IPC.

Activate the window mode for the operating software in the file "slguiconfig.ini".

### Procedure

1. Copy the "slguiconfig.ini" file from the folder /**siemens**/sinumerik/hmi/template/cfg.
2. Insert the copy into the folder /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg.
3. Open the file in the editor.
4. To activate or deactivate the window mode, make the following settings:
   - Activate window mode:
     In the section [WindowMode], ActivateWindowMode = true
   - Deactivate window mode:
     In the section [WindowMode], ActivateWindowMode = false
5. Restart SINUMERIK Operate.
   When the system has finished booting, the operating software is displayed in the window mode.

## 20.5 Setting text legibility by means of font smoothing

To ensure that particularly small TrueType fonts can be read more easily, the text display is already optimized under Windows by default. The settings perform fine font smoothing of the font and provide an improved display on the screen (Windows ClearType technology).

If you want to change the text legibility for SINUMERIK Operate, you can adjust the following parameters via the "slstartup.ini" file.

### Adapting the "startup.ini" file

| Parameter | Description | |
|---|---|---|
| `[FontSmoothing]` | Describes the section in which all of the parameters for setting the text legibility are located | |
| `UseSystemSettings` | Defines which system settings will be used for font smoothing and optimizing the display | |
| | `= true` | Windows settings |
| | `= false` | SINUMERIK Operate settings (default) |
| `UseFontSmoothing` | Activates font smoothing of the font type used in SINUMERIK Operate (only possible with parameters `UseSystemSettings` switched off) | |
| | `= true` | Switch on font smoothing |
| | `= false` | Switch off font smoothing (default) |

| Parameter | Description | |
|---|---|---|
| UseFontSmoothingType | Activates the Windows display optimization ClearType (only possible with switched off parameter UseSystemSettings and switched on parameter UseFontSmoothing) | |
| | = 1 | Only use font smoothing FE_FONTSMOOTHINGSTANDARD |
| | = 2 | Use ClearType technology FE_FONTSMOOTHINGCLEARTYPE |
| | = 0 | Switch off function (default) |

### Archive

The template of the file "slstartup.ini" for SINUMERIK Operate is stored in the following directory:

<Installation path>/siemens/sinumerik/hmi/cfg/slstartup.ini

Copy the file to one of the following directories:

<Installation path>/**user**/sinumerik/hmi/cfg

<Installation path>/**oem**/sinumerik/hmi/cfg

**Note**

If you want to achieve a better overview of the changes you have made yourself, simply delete the unchanged parameters from the file copy "slstartup.ini". The missing parameters continue to be read by the directory /siemens/sinumerik/hmi/cfg from the template.

## 20.6 Using interactive or silent installation versions

### General Information

If you install SINUMERIK Operate using PC/PCU, then the execution is realized using an MSI setup. The following installation versions are available, which you can use in the interactive or silent mode using the appropriate Windows command line:

- SINUMERIK Operate complete installation
- SINUMERIK Operate base installation with auxiliary files
- SINUMERIK Operate base installation without auxiliary files

### Execution

You control setup execution using command line arguments. The interactive mode /v allows you to transfer parameter strings to setup.exe. If you wish to run setup without user interaction, then you can extend the parameter string with the silent mode /qn.

Place the parameter string in inverted commas. If you use several parameters in the string, separate these using spaces.

If you do not specify any arguments when calling setup.exe, then the complete installation is run in the interactive mode.

**Note**

Do not put any spaces between `/v` and the parameter string.

**Note**

If you want a progress display in the silent mode, then instead of specifying `/qn` you can also specify `/qb!`.

## Interactive installation versions

| | |
|---|---|
| `setup.exe` | Complete installation in the interactive mode |
| `setup.exe /v"BASEONLY=1"` | Base installation in the interactive mode with help files |
| `setup.exe /v"BASEONLY=1 HELPFILES=0"` | Base installation in the interactive mode without help files |

## Silent installation versions

| | |
|---|---|
| `setup.exe /v"/qn"`<br>`setup.exe /s /v"/qn"` | Complete installation in the silent mode<br>(`/s` suppresses all of the setup messages) |
| `setup.exe /v"/qn BASEONLY=1"`<br>`setup.exe /s /v"/qn BASEONLY=1"` | Base installation in the silent mode with help files<br>(`/s` suppresses all of the setup messages) |
| `setup.exe /v"/`<br>`qn BASEONLY=1 HELPFILES=0"`<br>`setup.exe /s /v"/`<br>`qn BASEONLY=1 HELPFILES=0"` | Base installation in the silent mode without help files<br>(`/s` suppresses all of the setup messages) |

## Additional parameter assignments

### Installation path

If you install the PC version of SINUMERIK Operate, then you can additionally specify the installation path:
`setup.exe /s /v"/qn INSTALLDIR=C:\Programme\testdir"`

For spaces in the path:
`setup.exe /s /v"/qn INSTALLDIR=\"C:\Program Files\testdir\""`

### Exit code

You can identify a successful installation or errors based on the exit code of the setup.exe call:

| | |
|---|---|
| `Exit code == 0` | No error |
| `Exit code <> 0` | Error |

### Log file

During the installation you can generate a log file with `/L`:

```
setup.exe /s /v"/qb! /L*vx log.txt"
```

## 20.7 Setting the IP address of the NCU

From the user interface of the SINUMERIK Operate, you can set the IP address of the NCU with which it should connect.

### Procedure




1. Select the "Start-up" operating area.




2. Press the "HMI" softkey and the ">>" softkey.






Press the vertical softkey "NCU connection".
The "NCU Connection" window opens.
Enter the IP address of the NCU into the input field.




3. Press the "OK" softkey to confirm the data.

# 20.8 SINUMERIK Operate exiting

## Procedure




1. Press the <MENU SELECT> key on the operator panel.
   OR:
   Activate the <M> symbol on the touch panel.




External keyboard: Press the <F10> key.



2. Press the menu forward key.
   External keyboard: Press the <Shift> + <F9> keys.




3. Press the "EXIT" softkey.
   External keyboard: Press <F8>
   - on the PCU to terminate SINUMERIK Operate and the Windows system.
   - If you started SINUMERIK Operate operating software on the PCU in the service mode, then only the operating software is exited.
   - On the PC itself, only SINUMERIK Operate is exited.

# HT 8

# 21

## 21.1 Configuring the traversing keys

The traversing keys of the HT 8 are not labeled as their type of action is not predefined. The labeling of the traversing keys should adapt dynamically to the type of action of the keys. To recognize the type of action of the traversing keys, they are shown within the display in the vertical softkey area.

The menu of the traversing keys consists of two vertical rows each with eight keys so that up to 16 texts can be configured. The top and bottom traversing key row remain empty and can be assigned other functions.

The following data can be displayed:

- Machine axis name
- Alias name for machine axis
- Any language-dependent text
- Symbol

The following user-specific files are required. You can use the sample files as template:

| File | Meaning |
|---|---|
| "sljkconfig.ini" configuration file | File in which the traversing keys are configured. |
| Text file "oem_sljk_xxx.ts" | File for labeling traversing keys in a foreign language, xxx = language code |

### "sljkconfig.ini" configuration file

| Entries | Meaning | |
|---|---|---|
| [State_1] | Labeling type - changed via the PLC. | |
| ParamText_x_y | Text of the traversing key that is made up of parameters.<br>Two-line labeling is possible, whereby max. five characters are possible per line.<br>x: Specifies the position of the key within the row (2 to 7).<br>y: Specifies the key row (1 or 2). | |
| | %m1 | The machine axis name of the first axis is referenced and displayed as text. The current active axis number is read out from data block DB10. The name from the machine data is determined via this index. |
| | %n | Defines the position in the text for the line break. |
| | %a1 | The alias axis name of the first axis is referenced and displayed as text. The current active axis number is read out from data block DB10. The name from the "oem_sljk_eng.ts" text file is determined via this index. |

| Entries | Meaning |
|---|---|
| TextId_x_y | Text of the traversing key that is read from the text file (target language text). |
| Picture_x_y | File name of the icon to be displayed.<br>The files with the icons must be located in the following directories: /**oem**/sinumerik/hmi/ico and<br>depending on the resolution of the subdirectories:<br>/ico640<br>/ico800<br>/ico1024<br>/ico1280 |

## Procedure

1. Copy the "sljkconfig.ini" configuration file from the folder /**siemens**/sinumerik/hmi/template/cfg
2. Place the copy into the folder /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg
3. Open the copy in the Editor and define the traversing key labeling.

## Example of a "sljkconfig.ini" configuration file

12 axes are defined in the example, with text for softkey 1 and softkey 2, as well as an image on softkey 3 and softkey 4.

```
[Settings]
FileType = INI


; A few examples for further configuration options
; Alias names of machine-axes (%a1, %a2, etc.), from oem_sljk_deu.ts
(example)


[State_1]
TextId_1_1 = OEM_JK_TEXT_1
TextId_1_2 = OEM_JK_TEXT_2
ParamText_2_1 = %a1%n-
ParamText_2_2 = %a1%n+
ParamText_3_1 = %a2%n-
ParamText_3_2 = %a2%n+
ParamText_4_1 = %a3%n-
ParamText_4_2 = %a3%n+
ParamText_5_1 = %a4%n-
ParamText_5_2 = %a4%n+
```

```
ParamText_6_1 = %a5%n-
ParamText_6_2 = %a5%n+
ParamText_7_1 = %a6%n-
ParamText_7_2 = %a6%n+
Picture_8_1 = AlarmCancel.png
Picture_8_2 = AlarmNCReset.png
```

### "oem_sljk_eng.ts" text file

| Entries | Meaning |
|---|---|
| name | Freely selectable name of the text context.<br>In the text file template, the name of the text context is "SIJkLabels" and stands for the traversing key labeling (solution line jog key labels). This identifier is already stored in the configuration file. |
| source | Traversing key identifier of the respective axis. This text ID is referenced in the "sljkconfig.ini" configuration file with the "TextId_2_1".<br>The text IDs for the alias names (JK_AXISNAME_2 to JK_AXISNAME_7) must not be changed. |
| translation | Input of the foreign language text for the axis specified in <source>. |

### Language code in file names "xxx"

| Language | Language code |
|---|---|
| German | deu |
| English | eng |
| French | fra |
| Spanish | esp |
| Italian | ita |
| Chinese | chs |

### Procedure

1. You can copy the sample file " oem_sljk_deu.ts" from the following directory: /**siemens**/sinumerik/hmi/template/lng.
2. Store or create the file in the /**oem**/sinumerik/hmi/lng or /**user**/sinumerik/hmi/lng directory.
3. Give the file a name, e.g. for German texts: "sljk_deu.ts".
   If you wish to set-up the labeling for additional languages, a separate file must be created for each language. Save the file with the appropriate language code in the file name. To do this, use the language codes specified above.

4. Open the file and in the <message> and </message> area, define the labeling.
5. Restart the HMI.
   In order that the labeling is displayed during the program runtime, the file must be converted into a binary format. This conversion is only executed when the HMI powers up.

### Example of a text file "sljk_deu.ts"

12 axes are defined in the example, with text for softkey 1 (SF1) and softkey 2 (SF2):

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS>
<TS>
    <context>
       <name>SlJkLabels</name>
       <!-- Alias names for machine axis (%a1, %a2, etc.) -->
       <!-- Do not change the text ID (JK_AXISNAME_1, etc.) -->
       <message>
            <source>JK_AXISNAME_1</source>
            <translation>X</translation>
       </message>
       <message>
            <source>JK_AXISNAME_2</source>
            <translation>Y</translation>
       </message>
       <message>
            <source>JK_AXISNAME_3</source>
            <translation>Z</translation>
       </message>
       <message>
            <source>JK_AXISNAME_4</source>
            <translation>A</translation>
       </message>
       <message>
            <source>JK_AXISNAME_5</source>
            <translation>B</translation>
       </message>
       <message>
            <source>JK_AXISNAME_6</source>
            <translation>C</translation>
       </message>
       <message>
            <source>JK_AXISNAME_7</source>
            <translation>U</translation>
       </message>
```

```
        <message>
            <source>JK_AXISNAME_8</source>
            <translation>V</translation>
        </message>
        <message>
            <source>JK_AXISNAME_9</source>
            <translation>W</translation>
        </message>
        <message>
            <source>JK_AXISNAME_10</source>
            <translation>UV1</translation>
        </message>
        <message>
            <source>JK_AXISNAME_11</source>
            <translation>UV2</translation>
        </message>
        <message>
            <source>JK_AXISNAME_12</source>
            <translation>UV3</translation>
        </message>
        <!-- User defined language dependent text (example) -->
        <message>
            <source>OEM_JK_TEXT_1</source>
            <translation>SF1</translation>
        </message>
        <message>
            <source>OEM_JK_TEXT_2</source>
            <translation>SF2</translation>
        </message>
</TS>
```

### See also

Supported languages (Page 502)

## 21.2 Configuring user-specific key labeling

### Labeling the keys for the HT 8

The texts of the CPF menu (CPF: Control Panel Function) for the HT 8 can be labeled with your own texts in the particular language of the country.

The texts are created in the "slck_xxx.ts" file.
"xxx" = language code for the corresponding language-specific labeling.

You can create and edit this file using the HMI or also externally on a PC.

**Note**

If you wish to create or edit the file on a PC, use an editor that supports UTF-8 coding.

## Language code in file names "xxx"

| Language | Language code |
|---|---|
| German | deu |
| English | eng |
| French | fra |
| Spanish | esp |
| Italian | ita |
| Chinese | chs |

## Creating key labeling

| Tag | Meaning |
|---|---|
| source | Designation for the user softkey. "SK_USERKEY1" to "SK_USERKEY16" are possible, whereby the names may not be changed. |
| comment | User-specific description of the key assignment. |
| translation | Text that should be shown on the key.<br>• A maximum of 10 characters are possible per line.<br>• 2-line labeling is possible, whereby the line break is set using "%n". |
| remark | Remark for key assignment. |
| chars | Number of characters. A maximum of 10 characters per line are possible. |
| lines | Number of lines. 2 lines are possible. |

## Procedure

1. You can copy the sample file " oem_slck_deu.ts" from the following directory: /**siemens**/ sinumerik/hmi/template/lng.
2. Store or create the file in the /**oem**/sinumerik/hmi/lng or /**user**/sinumerik/hmi/lng directory.
3. Give the file a name, e.g. for German texts: "slck_deu.ts".
   If you wish to create the key labeling for additional languages, then a separate file must be created for each language. Save the file with the appropriate language code in the file name. To do this, use the language codes specified above.

4. Open the file and in the <message> and </message> area, define the key labeling.
5. Restart the HMI.
   In order that the key labeling is displayed during the program runtime, the file must be converted into a binary format. This conversion is only executed when the HMI powers up.

## Example of a key label

```
<!DOCTYPE TS><TS>
<context>
   <name>SlCkDialog</name
   <message>
     <source>SK_USERKEY1</source>
     <comment></comment>
     <translation>U1</translation>
     <remark>User key 1</remark>
     <chars>10</chars>
     <lines>2</lines>
     <languageIndependent>true</languageIndependent>
   </message>
   <message>
     <source>SK_USERKEY2</source>
     <comment></comment>
     <translation>U2</translation>
     <remark>User key 2</remark>
     <chars>10</chars>
     <lines>2</lines>
     <languageIndependent>true</languageIndependent>
   </message>
   <message>
   .....
   </message>
</context>
</TS>
```

## See also

Range of alarms (Page 184)

# 21.3 Configuring the function display at user-specific keys (U keys)

## Function

Active functions can be displayed at the configurable user keys via the PLC. For instance, small LEDs can be emulated on the softkeys.

You configure the function in the "slckcpf.ini" file.

## Interface signals

The PLC bits are in the output image of the PLC-HT 8 interface and are analog to those in the input image.

| Signals to the MCP1 (or MCP2)<br>Interface PLC → HT 8 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| AB n + 1 | | U4 | U3 | | U2 | U1 | | |
| AB n + 4 | U9 | U10 | U11 | U12 | U13 | U14 | U15 | U16 |
| AB n + 5 | | U8 | U7 | U6 | U5 | | | |

## Adapt the display in the "slckcpf.ini" file

| Section | Description | |
|---|---|---|
| `UserKeyLEDIcon` | Name of the icon file.<br>Standard entry: `led_green.png` | |
| | `OFF` | Deactivates the icon display. |
| | `PRESSED` | Displays the active functions by pressing down the softkey. |
| `UserKeyLEDIconAlignment` | Specifies the position of the icon.<br>Standard position: `AlignLeft \| AlignTop`<br>Horizontal and vertical alignments can be combined. The two names are separated by the "`\|`" character.<br>The following alignments are possible: | |
| | `AlignLeft` | Left |
| | `AlignRight` | Right |
| | `AlignHCenter` | Horizontal, center |
| | `AlignTop` | Top |
| | `AlignBottom` | Bottom |
| | `AlignVCenter` | Vertical, center |
| `UserKeyLEDMap` | Specifies the start address of the output image.<br>Entries can be made in the following form: "`DBx.DBBy`", "`ABx`", "`MBx`".<br>Default setting: The start address is determined using DB7 MCP1Out (or MCP2Out). | |

| Section | Description | |
|---|---|---|
| U1LED ... U16LED | The status bit address can be defined - different from the output image - using these entries.<br>Note: The offsets from the HT 8 output image, for one or several softkeys, are not taken into consideration. | |
| VarIncLED | | |
| SBLLED | | |
| WCSLED | | |
| showVarIncLED | true | When the increment mode is active, the icon is also displayed on the "[VAR]" softkey. |
| | false | The icon is not displayed. |
| showSBLLED | true | When the SingleBlock mode is active, then the icon is also displayed on the "Single Block" softkey. |
| | false | The icon is not displayed. |
| showWCSLED | true | When the SingleBlock mode is active, then the icon is also displayed on the "Single Block" softkey. |
| | false | The icon is not displayed. |

1. You can copy the sample file "slckcpf.ini" from the following directory: /**siemens**/sinumerik/hmi/template/cfg
2. Store the file in the /**oem**/sinumerik/hmi/cfg or /**user**/sinumerik/hmi/cfg directory.
3. If you use your own icon, then locate it together with the corresponding resolution for HT 8 directory: /**oem**/sinumerik/hmi/ico/ico640 or /**user**/sinumerik/hmi/ico/ico640.
4. Open the file and make the appropriate settings.

## Sample file "slckcpf.ini"

```
Template for the configuration of the
; HT8 control panel function menu
;
; To activate the settings remove the
; commentary ';' at the beginning of the line
;; Display settings of the user key softkey leds
[UserKeyLED]
; Filename of the LED icon
;UserKeyLEDIcon = led_green.png
; Alignment of the LED icon
;UserKeyLEDIconAlignment= AlignLeft | AlignTop


; Use following led map start address instead of calculating
DB7.MCP1Out
;UserKeyLEDMap = AB0
```

```
; Use the following settings to use this status bits instead of the
led map for a specific sk
;U1LED=/channel/parameter/R[U1,1]
;....
;U16LED=/channel/parameter/R[U1,16]
;VarIncLED = DB11.DBX8.5
;SBLLED = DB21.DBX0.4
;WCSLED = DB19.DBX0.7

; Show a LED for the var inc sk
;showVarIncLED = true
; Show a LED for the var single block sk
;showSBLLED = true
; Show a LED for the var wcs/mcs sk
;showWCSLED = true
```

# List of abbreviations/acronyms

# A

| Abbreviation | Meaning |
|---|---|
| ASCII | American Standard Code for Information Interchange:<br>American coding standard for the exchange of information |
| BAG | Mode group |
| OPI | Operator Panel Interface |
| CNC | Computerized Numerical Control: Computerized numerical control |
| CEC | Cross Error Compensation: Sag compensation |
| CP | Communication Processor |
| DCP | Discovery and basic Configuration Protocol: ProfiNet protocol |
| DHCP | Dynamic Host Configuration Protocol |
| DIN | Deutsche Industrie Norm |
| DNS | Domain Name System |
| DIR | Directory: Directory |
| DRAM | Dynamic Random Access Memory |
| DRF | Differential Resolver Function: Differential resolver function (handwheel) |
| EES | Execution from External Storage |
| ERTEC | Enhanced Real Time Ethernet Controller |
| ESR | Extended Stop and Retract |
| ETC | ETC key ">"; Softkey bar extension in the same menu |
| FRAME | Data block (FRAME) |
| FIFO | First In - First Out: Method of storing/retrieving data in/from a memory. |
| GDIR | Global Directory: Global part program memory |
| GUD | Global User Data: Global user data |
| GUI | Graphical User Interface |
| MSD | Main Spindle Drive |
| HSC | High Speed Cutting: High-speed machining |
| INC | Increment: Incremental dimension |
| INI | Initializing Data: Initializing data |
| IP | Internet Protocol |
| IPO | Interpolator |
| ISO | International Standard Organization |
| JOG | Jogging: Setup mode |
| LED | Light Emitting Diode: Light emitting diode |
| LLDP | Link Layer Discovery Protocol |
| LUD | Local User Data: Local user data |
| MAC | Media Access Control |
| MB | Megabyte |
| MD | Machine data |
| MDA | Manual Data Automatic: Manual input |

| Abbreviation | Meaning |
|---|---|
| MCS | Machine Coordinate System |
| MLFB | Maschinenlesbare Fabrikatbezeichnung |
| MPF | Main Program File: NC part program (main program) |
| MCP | Machine Control Panel: Machine control panel |
| NC | Numerical Control: Numerical control |
| NCK | Numerical Control Kernel: NC kernel with block preparation, traversing range, etc. |
| NCU | Numerical Control Unit: NCK hardware unit |
| NFS | Network File System |
| NTP | Network Time Protocol |
| ZO | Zero offset |
| OEM | Original Equipment Manufacturer |
| OID | Object Identifier: Information identifier |
| OP | Operator Panel |
| PCU | Programmable Control Unit |
| PI service | Program Invocation Services |
| PG | Programming device |
| PLC | Programmable Logic Control |
| RCS | Remote Control System |
| RFC | Request for Comments |
| REF | REFerence point approach function |
| REPOS | REPOSition function |
| ROV | Rapid Override: Input correction |
| RPA | R-Parameter Active: Memory area in NCK for R- NCK for R parameter numbers |
| SBL | Single Block: Single block |
| SD | Setting Data |
| SDB | System Data Block |
| SEA | Setting Data Active: Identifier (file type) for setting data |
| SK | Softkey |
| SMB | Server Message Block |
| SNMP | Simple Network Management Protocol |
| SPF | Sub Program File: Subroutines |
| SRAM | Static Random Access Memory: Static RAM (non-volatile) |
| SSH | Secure Shell: Network protocol |
| SW | Software |
| SYF | System Files: System files |
| TCU | Thin Client Unit |
| TEA | Testing Data Active: Identifier for machine data |
| TMA | Tool Magazine Active: Magazine data |
| TO | Tool Offset: Tool offset |
| TOA | Tool Offset Active: Identifier (file type) for tool offsets |
| UDP | User Datagram Protocol: Network protocol |
| UFR | User frame |
| VNC | Virtual Network Computing |

| Abbreviation | Meaning |
|---|---|
| FD | Feed Drive |
| WCS | Workpiece Coordinate System |
| TM | Tool Management |

# Index

## 3

3D simulation, 139, 143

## A

Access levels, 39
Access MyMachine, 14, 23
Access right
  Effect, 43
  Global, 42
  individual, 42
  Remote access, 42
Action log
  actlog.ini, 291
  Log file, 289
  Open log file, 287
  Saving log file, 289
  Search for events, 288
  Setting the log size, 291
  Settings, 285
action.com, 289
action.log, 289
Activate keyboard, 31
Activating/deactivating window mode, 523
actlog.ini, 291
Actual value window
  Setting font size, 119
actual_actionlog.com, 285
actual_crashlog.com, 285
Adapt tool measurement
  Examples when milling, 488
  for turning, 493
  when milling, 487
Adapt workpiece measurement, 482
Additional language
  installing, 501
  Uninstalling, 502
Alarm log
  Filter properties, 189
  Settings, 188
Alarm number ranges, 184
Alarm texts
  Changing colors, 176
  Create file, 168
  Creating, 165, 168
  Creating foreign-language texts, 167
  Deactivating a warning, 197
  Registering the file, 169
  Replacing, 180
  Replacing standard alarm texts, 180
  Select font color, 165
  Standard parameter specification, 185
  Supported languages, 502
alarmtext_conversion.log, 170, 172, 175, 183, 186
Analog spindle, 150
Archive
  create for Hotline, 208
  Creating, 203
  Line checksum, 200
  Login data (credentials) handling, 34
  Reading in, 204
AS-i
  Components, 324
  Configure diagnostics, 326
  Diagnostics, 324
Axis diagnostics, 259
Axis-specific machine data, 97
Axis-specific setting data, 100

## B

Backing up setup data, 212

## C

Changing the design, 32
Channel menu
  Configuring, 87
  Configuring operator stations, 90
  Configuring the total structure, 89
  Defining channel groups, 89
Channel operating message
  Standard configuration, 124
Channel operational message, 123
Channel-specific machine data, 97
Channel-specific setting data, 100
Circular position pattern, 365
Circumferential groove, 365
Clamping
  Setting up function, 145
Comment input
  ini file, 23
  XML file, 22
Commissioning the drive system, 163

config.ini, 91
Configuring OEM icons, 136
Configuring OEM texts, 135
Configuring operator stations, 90
Contour grooving, 385
Contour milling, 364
Contour turning, 385
Coordinate system
MD52000, 351
crash.com, 289
crash.log, 289
Creating online help
Creating a directory, 340
HTML_Creating file, 337
Replacing the standard online help, 339
XML_Creating file, 339
CSV file, 281
CUST_800.SPF, 356
CUST_832.SPF, 356, 460
CUST_CLAMP, 357
CUST_M6, 357, 359
CUST_MEACYC
Structogram, 468
CUST_MEACYC.SPF, 356
CUST_MEAPROT.SPF, 356, 496
CUST_MULTICHAN, 357, 363
CUST_T, 357, 359
CUST_TECHCYC.SPF, 356
Aligning, 361
Cycle alarms, 413
CYCLE495, 403
CYCLE60, 365
CYCLE63, 364
CYCLE64, 364
CYCLE79, 365
CYCLE800
Structogram, 437
CYCLE832
Advanced Surface, 458
Top Surface, 462
CYCLE84, 355
CYCLE840, 355
CYCLE930, 385
CYCLE950, 385
CYCLE951, 385
CYCLE952, 385
CYCLE99, 385
CYCLE996, 442
Cycles version
Define, 499
Displaying, 498
Cylinder surface transformation
in turning, 390
Milling, examples, 370
under milling, 368
under ShopTurn, 392
with groove side offset, 393
without groove side offset, 391
Cylindrical grinding, 402
Displaying softkeys, 350

## D

Data backup, 199
Date
Selecting the format, 26
Setting, 27
Deactivate keyboard, 31
dealer.xml, 276
Defining channel groups, 89
DHCP server detection, 242
Diagnostics
AS-i, 326
Ethernet/PROFINET network, 242
HMI trace, 292
Direction of rotation
Counterspindle, 382
Main spindle, 376
Directory structure, 17
Display filter
Machine data, 104
Display groups, 105
Display machine data, 100
Search, 108
Display Manager
Activating, 57
Display configurations, 56
Displaying AS-i, 326
Drilling, 354
Drive
Global part program memory, 85
Logical drive, 83
Set EXTCALL, 84
Setting up, 83
Drive parameters
Display/change, 102
for machine data, 97
Drive system
Diagnostics, 296
Diagnostics - details, 297
Drive-specific machine data, 97

## E

Editing an xml file, 22
Editing enum values, 107
Engraving, 365
Entering BICO values, 107
Error analysis, 230
Error file of the alarms, 186
Ethernet/PROFINET
  Diagnostics, 242
  Displaying messages, 243
  Network trace, 244
  Trace parameters, 245
EXIT, 527
  Displaying the softkey, 31
Expert parameters, 105
EXTCALL
  switch off and on again, 84

## F

Face machining
  in turning, 394
  under ShopTurn, 396
fast block search, 129
File
  Copy, 20
  Copying with WinSCP, 23
  Cut, 21
  Delete, 21
  Displaying in preview window, 22
  External editing, 22
  insert, 20
  Open, 20
  Renaming, 21
Font smoothing, 523
Form-truing, 403

## G

G code groups
  Configuring the display, 121
G group 59, 459
General machine data, 97
General setting data, 100
Generating a crashlog file, 286

## H

Hardware components
  Adding, 281
  Entering, 280
HEAD_1, 425
High-Speed Settings, 458
Hirth joint, 409, 453
HMI PRO RT, 322
HMI restart
  Displaying the softkey, 31
  execution, 32
HMI trace, 292
HOLES2, 365
HT 8
  Activating, deactivating the virtual keyboard, 31
  Configuring the traversing keys, 529
  Key labeling, 533
  LED display on user keys, 536
  oem_sljk_deu.ts, 531
  slguiconfig.ini, 31
  sljkconfig.ini, 529

## I

identSNAPSHOT, 273
Inclined Y axis
  in turning, 397
  under ShopTurn, 398
Index text file for alarms
  Create file, 170
  Creating texts, 171
  Registering, 171
Inserting a run up screen, 33
Installation
  Interactive, 525
  silent, 525

## K

Keyboard
  Setting, 30
Kinematics
  Check list, 413
  Examples, commissioning, 425
  measured, 442
  with Hirth joint, 409

## L

Language
Defining the selection mode, 25
Language code, 502, 503
License key
Backing up from CompactFlash card, 80
Entering, 80
Reading in, 80
License requirement
Exporting, 79
Licenses, 75
Log file
Action log, 289
Logbook, 282
Documenting startups, 283
Entry search, 285
Import templates, 278
Machine identity, 273
Making an entry, 284
Output, 279
Logging the measurements
Settings for AUTOMATIC, 495
Logging the measuring
Settings for JOG, 495

## M

ma_types.xml, 277
Machine data
Axis selection, 99
Configuring plain texts, 116
Display filter, 104
Displaying, 97
Editing enum values, 107
Editing hexadecimal values, 106
Entering BICO values, 107
Information, 98
Overview, 97
Search, 108
Machine identity, 273
Machine operating area, 119
Machine-specific information, 279
Create a template, 275
Entering, 274
Machining time recording
Activating, 137
Managing data, 110
Manual machine
Activating the function, 131
Parameter assignment, 131
Message texts
Creating, 173
creating for several channels, 174
Part program, 173
Milling
Cylinder surface transformation, 368
Direction of rotation for position pattern, 366
Displaying softkeys, 350
Examples, setting-up a milling machine, 370
Settings under ShopMill, 366
Tool measurement, 474
Turning on milling machines, 444
Workpiece measurement, 472, 485
Multiple clamping of different workpieces (option), 373
Multiple edge, 365

## N

NC/PLC variables
Changing, 253
displaying, 251
NCU connection
setting - under PCU, 526
setting - under Windows, 526
netnames.ini, 90
Network
Configuring, 217
Network adapter, 228
Configuring, 238
Network diagnostics, 227
Number ranges of alarms, 184

## O

oem.xml, 277
oem_ae_database_conversion.log, 179
oem_alarms_eng.ts, 168
oem_indextexts_eng.ts, 170
oem_slaedatabase.xml, 176
oem_slaesvcadapconf.xml, 169
oem_slck_deu.ts, 534
oem_sljk_deu.ts, 531
oem_text_conversion.log, 170, 186
OEMFrame application
configuring, 505
extended parameterization, 513

FindWindow program application, 522
integrating, 505
oemsubs.xml, 277
Offset
Configuring, 136
Operating state, 256
Operator panels
Configuring, 224
Orientable tool holder, 494

## P

parallel user interfaces, 33
Password
Changing, 40
Delete, 41
Set, 40
Performing a start-up
Reading in original status, 207
PLC hardware upgrade archive
Creating, 204
Probe
Example of a test program, 464
Testing the function, 464
to measure the tool, 463
to measure the workpiece, 463
PROFIBUS
Connections, 293
DP slaves, 295
PROFIBUS/PROFINET diagnostics, 322, 323
PROG_EVENT.SPF, 293, 357
Extend, 358
Program access
setting via protection level, 47
Program list
Configuring a softkey, 48
Creating, 48
Program test
activate, 125
deactivate, 125
Programming G code programs, 131

## R

RCS Commander, 14
Remote diagnostics
Adapting, 299
Replacing standard alarm texts
Create file, 180
Creating texts, 180
Registering the alarm text file, 181
Replacing the logo, 120
Reset
Extended, 103
Rotary axis vectors
V1, 414
V2, 414

## S

Screen
darken screen, 32
saver, 32
Time until the screen is darkened, 32
Screenshots
Copy, 273
Creating, 272
Open, 273
Search
Display machine data, 108
Machine data, 108
Setting data, 108
Search mode
Configuring, 128
Searching
Logbook entry, 285
Service drive
Display, 264
Parameter description, 265
Service overview, 256
Axes and drives, 258
Service Platform Industry, 273
Setting data
Axis selection, 102
Channel selection, 102
Displaying, 100
Overview, 97
Search, 108
Setting up a counterspindle
ShopTurn, 381
Setting-up a lathe with milling tools, 389
Setting-up measuring in JOG, 470
ShopMill
Settings for milling, 366
ShopTurn
Cylinder surface transformation, 392
Face machining, 396
Inclined Y axis, 398
Setting up a counterspindle, 381
Settings for turning, 376
Sidescreen
activating, 50
Display ABC keyboard, 51

Display machine control panel, 52
Display translations, 55
In-house configuration, 53
Simulation
Cycle application, 357
Deactivate, 142
General conditions, 139
Grinding technology, 141
Milling technology, 141
Tool change time, 143
Turning technology, 141
Simultaneous recording, 143
Activating the machining time, 144
Option, 143
slaesvcadapconf.xml, 169, 171, 174, 181, 197
slaesvcconf.xml, 175, 183
slckcpf.ini, 537
slhlp.xml, 335
slmagcodeconfig.ini, 121
SLOT2, 365
sltlprodiaghmi.ini, 324, 325
Softkey
Access level, 46
Original setting, 46
Source ID, 185
Source URL, 185
Specify the IP address of the NCU, 526
Spindle analog, 150
Spindle control, 149
Spindle diagnostics, 151
Additional temperature sensor, 155
Clamping state, 157
Clamping time statistics, 161
Diagnostic statistics, 160
Logistics data, 161
Motor temperature sensor, 155
Speed limitation, 159
Spindle load, 156
Temperature, 153
Temperature histograms, 156
Standard parameter specifications of alarm texts, 185
Starting SINUMERIK Operate
IPC, 13
NCU, 13
PCU, 13
under Windows, 13
Start-up archive
Creating, 200, 213
Reading in, 204
STAT, 125
Station-related network diagnostics, 227
Status display
Inserting a logo, 120
Status display with icons, 133
Stock removal, corner, 385
Surface grinding, 402
Displaying softkeys, 350
Swiveling
activating, 405
Back up data set, 424
Commissioning machine kinematics, 414
Configuring input screens, 405
Creating angled tools, 408
Declare data sets, 407
Example HEAD_2, 426
Example MIXED_45, 429
Example TABLE_45, 427
Example TABLE_5, 431
Retraction, 423
Rotary axes, 420
Swiveling:
Offset vectors, 416
System utilization, 271


## T


Tailstock
Option, 143
Tapping, 355
TEACH IN, 127
Technology
Cylindrical grinding, 402
Drilling, 354
Milling, 364
Surface grinding, 402
Swiveling, 404
Turning, 385
Template files, 20
Test license, 78
Text legibility, 523
TextPad, 14
Thread-cutting, 385
Time
Selecting the format, 27
Setting, 27
Time server, 29
Time zone, 30
Tool clamping angle, 452
Tool measurement
under milling, 474
under turning, 479
Topologies, 247

TRAANG, 399
in turning, 397
under ShopTurn, 398
Trace
Changing the scale, 318
Creating a session, 305
Cursor A/B positioning, 320
Data acquisition, 313
Data storage on CU, 314
Define zoom area, 320
Display peak-to-peak value, 321
Display peak-to-valley value, 321
Displaying details of the variables, 312
Drive variables, 308
Editing a session, 309
Editing the graphic display, 316
Filtering variables, 307
Insert variable, 308
Load session, 307
Overview, 303
PLC/NC/servo variables, 307
Replace variable, 309
Save options, 315
Save session, 306
Searching for variables, 307
Selecting a variable, 308, 318
Sequence, 304
Session, 308
Setting grid lines, 315
Settings (drive), 313
Settings (PLC, NC, servo), 313
Start recording, 316
Starting a session, 307
Stop recording, 316
Variable attributes, 309
Zooming characteristic graphs, 319
TRACON, 399
TRACYL, 400
in turning, 390
under ShopTurn, 392
Transformation
Settings for milling, 371
TRANSMIT
in turning, 394
under ShopTurn, 396
with Y axis, 395
TU, 125
Turning
Cylinder surface transformation, 390
Displaying softkeys, 350
Examples, setting-up a lathe, 390
Face machining, 394
Inclined Y axis, 397
Settings under ShopTurn, 376
Speed limitation, 400
Technology cycles, 385
Tool measurement, 479
Workpiece measurement, 492
Turning on milling machines
Behavior at power on, 452
Behavior at reset, 451
Block search, 457
CUST_800, 454
Hirth joint, 453
Kinematics transformation, 449
Setting the coordinate system, 454
Setting up, 444
Tool clamping angle, 452

## U

USB FlashDrive, 14
USB keyboard, 14
User view
Creating, 113
Delete, 116
Editing, 114
Importing, 112
user.xml, 277
Using angular tools, 408

## V

Variable screen forms, 255
VNC operating area, 35
Setting up, 38

## W

Window mode, 523
WinSCP, 14, 23
Workpiece measurement
under milling, 472, 485
under turning, 492

SIEMENS

Generating user dialogs 1

Generating commissioning dialogs 2

# SINUMERIK

# SINUMERIK 840D sl Easy XML

Programming Manual

Valid for:

CNC software Version 4.8 SP3

08/2018
6FC5397-1DP40-6BA2

## 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

Document order number: 6FC5397-1DP40-6BA2
℗ 08/2018 Subject to change

Copyright © Siemens AG 2014 - 2018.
All rights reserved

# Table of contents


**1 Generating user dialogs....5**

1.1 Scope of functions....5

1.2 Fundamentals of Configuration....7

1.3 Configuration files....11

1.4 Structure of configuration file....14

1.5 Language dependency....20

1.6 XML diagnostics....21

1.7 XML identifier....23
1.7.1 General structure....23
1.7.2 Instruction/identifier descriptions....24
1.7.3 Color coding....48
1.7.4 Special XML syntax....48
1.7.5 Operators....49
1.7.6 System variables....50
1.7.7 Generating softkey menus and dialog forms....51

1.8 Generating user menus....90
1.8.1 Creating processing cycle forms....90
1.8.2 Substitution characters....93

1.9 Addressing components....94
1.9.1 PLC addressing....94
1.9.2 Addressing NC variables....95
1.9.3 Channel-specific addressing....95
1.9.4 Generating NC/PLC addresses during the runtime....95
1.9.5 Addressing drive components....96
1.9.6 Example: Determine the DO number for the Motor Module....98
1.9.7 Addressing machine and setting data....104
1.9.8 Channel-specific machine data....105
1.9.9 Addressing user data....106

1.10 Predefined functions....107

1.11 Multitouch operation....153
1.11.1 Multitouch function....153
1.11.2 Programming finger gestures....155
1.11.3 Gesture control for graphics....156
1.11.4 Gesture processing....159

1.12 Configuring your own buttons....161
1.12.1 Pushbutton....161
1.12.2 Functions of the pushbutton....163
1.12.2.1 Sub-tags for the pushbutton....163
1.12.2.2 Properties for the pushbutton....165
1.12.2.3 Control variables for the pushbutton....167

1.12.3 Switch on/off....171
1.12.4 Functions of the switch....172
1.12.4.1 Properties for the switch....172
1.12.4.2 Control variables for the switch....173
1.12.5 Radio button....175
1.12.6 Checkbox....175
1.12.7 Groupbox....176
1.12.8 Scroll area....177

1.13 Sidescreen application....180
1.13.1 Easy XML in the Sidescreen....180
1.13.2 Integrating Sidescreen dialogs....181
1.13.3 Language and text management....182
1.13.4 Sidescreen components....183
1.13.4.1 Sidescreen element....183
1.13.4.2 Sidescreen widget....184
1.13.4.3 Sidescreen page....186

1.14 Dialog selection via PLC hardkeys....188

1.15 Assigning user dialogs to reserved softkeys....191
1.15.1 Defining language-neutral softkey text....192
1.15.2 Assigning an Easy XML area....193
1.15.3 Tag descriptions....197

1.16 Creating language-neutral texts....199

1.17 Project-specific text files....200
1.17.1 Creating project-specific text files....200
1.17.2 Specifying text context....202

**2 Generating commissioning dialogs....205**

2.1 Overview of functions....205

2.2 Configuration in the PLC user program....207

2.3 Display on the user interface....210

2.4 Creating language-dependent texts....211

2.5 User example for a power unit....212

2.6 Script language....214
2.6.1 CONTROL_RESET....217
2.6.2 FILE....217
2.6.3 OPTION_MD....218
2.6.4 PLC_INTERFACE....219
2.6.5 POWER_OFF....220
2.6.6 WAITING....220
2.6.7 XML identifiers for the dialog....220
2.6.8 SOFTKEY_OK, SOFTKEY_CANCEL....221

**Index....223**

# Generating user dialogs

# 1

## 1.1 Scope of functions

### Overview

The "Generate user dialogs" function offers an open structure and enables the user to develop customer-specific and application-specific user interfaces in SINUMERIK Operate.

The control system offers an XML-based script language for generating user dialogs.

This script language makes it possible to display machine-specific menus and dialog forms in the <CUSTOM> operating area in SINUMERIK Operate.

All dialog forms can be designed on a language-neutral basis. In such cases, the system reads out the texts to be displayed from the accompanying language database.

### Use

The defined XML instructions offer the following properties:

1. Display dialogs containing the following elements:
   - Softkeys
   - Variables
   - Texts and Help texts
   - Graphics and Help displays
2. Call dialogs by:
   - Pressing the (start) softkeys
3. Restructure dialogs dynamically:
   - Edit and delete softkeys
   - Define and design variable fields
   - Insert, exchange, and delete display texts (language-dependent or language-neutral)
   - Insert, exchange, and delete graphics
4. Initiate operations in response to the following actions:
   - Displaying dialogs
   - Inputting values (variables)
   - Selecting a softkey
   - Exiting dialogs
5. Data exchange between dialogs

6. Variables
   - Read (NC, PLC and user variables)
   - Write (NC, PLC and user variables)
   - Combine with mathematical, comparison or logic operators
7. Execute functions:
   - Subprograms
   - File functions
   - PI services
8. Apply protection levels according to user classes

The valid elements (tags) for the script language are described in the "XML tags" (Page 23) section.

**Note**

The following section "Basic principles of configuration" is not intended as a comprehensive description of XML (Extensible Markup Language). Please refer to the relevant specialist literature for additional information.

# 1.2 Fundamentals of Configuration

## Configuration files

The defining data for new user interfaces are stored in configuration files. These files are automatically interpreted and the result displayed on the screen. Configuration files are not stored in the software supplied and must first be set up and loaded by the user.

An XML editor or another form of text editor can be used to generate the configuration files.

**Note**

File names may only contain lowercase letters.

## Menu tree principle

Several interlinked dialogs create a menu tree. A link exists if you can switch from one dialog to another. You can use the newly defined horizontal/vertical softkeys in this dialog to call the preceding or any other dialog.

Configured start softkeys can be used to create a further menu tree behind the start menu:

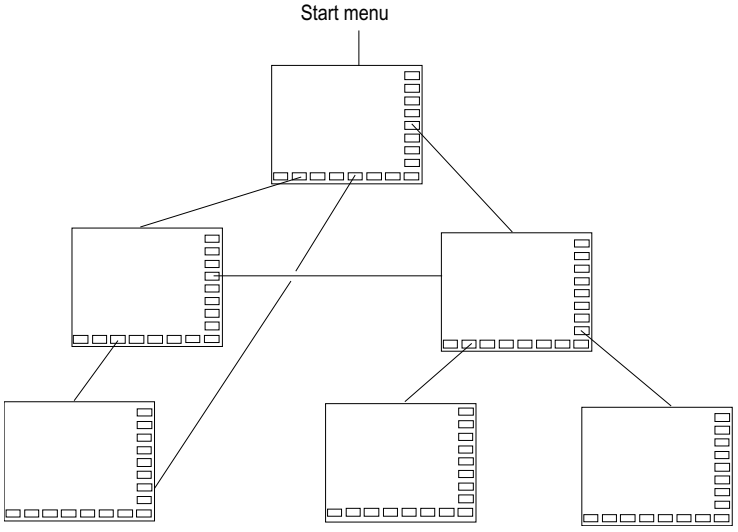



Figure 1-1 Menu tree for user dialogs

## Start menu

The start menu is defined by the name "main" in the "xmldial.xml" file. The start menu is used to initiate your own operating sequences.

You can use the "main" menu to link your own dialogs or additional softkey bars so that they can be loaded and used for executing additional actions.

The figure below shows the manufacturer’s folder "System CF-Card/oem/sinumerik/hmi" on the control.

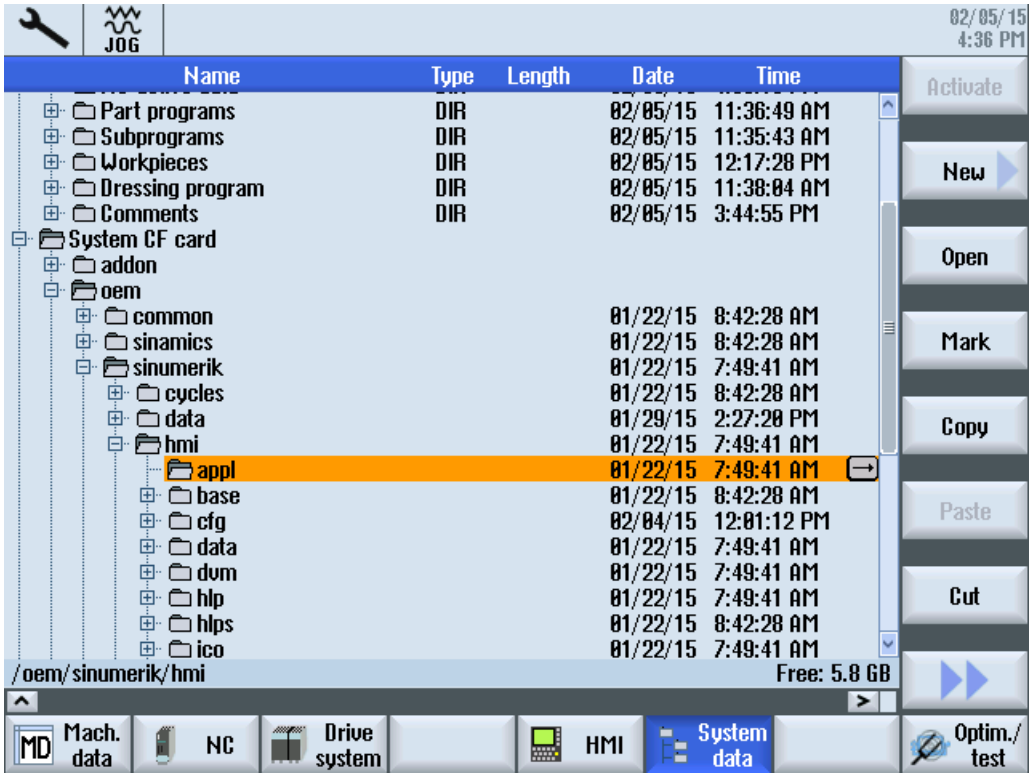



Figure 1-2 Manufacturer's folder

The following files in the manufacturer's folder "System CF-Card/oem/sinumerik/hmi" on the control are needed to configure user dialogs:

Table 1-1 Files for configuration

| File type | Name of the file | Description | Storage location in operating area Set-up > System data |
|---|---|---|---|
| Script file | "xmldial.xml" | This script file uses XML tags to control how the configured soft-key menus and dialog forms in SINUMERIK Operate will appear in the <CUSTOM> operating area. | Manufacturer's folder > subdirectory "appl" for the applications |
| Text file | "oem_xml_screens_xxx.ts" | This text file contains the texts for the menus and dialog forms for the individual languages. | Manufacturer's folder > subdirectory "lng" for the desired languages |
| Bitmaps | | The control system supports BMP and PNG formats. | Manufacturer's folder > subdirectory "ico" |
| XML files inserted in the "xmldial.xml" control file with the "INCLUDE" XML tag. | E.g. "machine_settings.xml" | These files also contain programmed instructions for displaying the dialog forms and parameters in SINUMERIK Operate. | Manufacturer's folder > subdirectory "appl" for the applications |

## Easy XML connection points

The following connection points are available for Easy XML scripts:

| Hardkey | Connection point |
|---|---|
| Softkey operating menu | Standard script file name "xmldial.xml"<br>Activated in the file "slamconfig.ini"<br><br>**Example**<br>`[CustomXML]`<br>`TextId=SL_AM_CUSTOM`<br>`SidescreenTextId=SL_SIDESCREEN_CUSTOM`<br>`TextFile=slam`<br>`TextContext=SlAmAreaMenu`<br>`SoftkeyPosition=12`<br>`Visible=true` |
| Menu User | Standard script file name "menu_user.xml" |
| Menu Function | Standard script file name "menu_function.xml" |
| PLC hardkey | The script file name relates to the key description.<br><br>**Example**<br>`KEY50.0 = area:=CustomXML, dialog:=SlEECustomDialog, cmdline:="-conf slagmdialog.hmi -mainModule restore.xml -entry cmc2"`<br>`KEY51.0 = area:=CustomXML, dialog:=SlEECustomDialog, cmdline:="-conf slagmdialog.hmi -mainModule activate.xml -entry cmc1"` |
| MMC command | The script file name relates to the NC command description.<br><br>**Example**<br>`MMC("XML, XML_ON, TEST.XML, cmd1","A")` |

| Hardkey | Connection point |
|---|---|
| Reserved softkeys of a working area | The script file name relates to the menu description.<br><br>**Example**<br>`<MENU name="DgGlobalHu">`<br>`<ETCLEVEL id="0">`<br>`<SOFTKEYGROUP name="GroupEtc">`<br>`<SOFTKEY position="7">`<br>`<PROPERTY name="textID"`<br>`type="QString">DG_SK</PROPERTY>`<br>`<PROPERTY name="translationContext"`<br>`type="QString">EASY_XML</PROPERTY>`<br>`<FUNCTION args="-area CustomXML -`<br>`dialog SlEECustomDialog -mainModule`<br>`dg.xml -entry main"`<br>`name="switchToDialog"/>`<br>`</SOFTKEY>`<br>`</SOFTKEYGROUP>`<br>`</ETCLEVEL>`<br>`</MENU>` |
| Sidescreen | The script file name relates to the sidescreen description of the file "slsidescreen.ini". |
| Easy Extend softkey | Standard script file name "agm.xml" |

# 1.3 Configuration files

## Introduction

The figure below shows the manufacturer's folder "System CF-Card/oem/sinumerik/hmi" on the control.

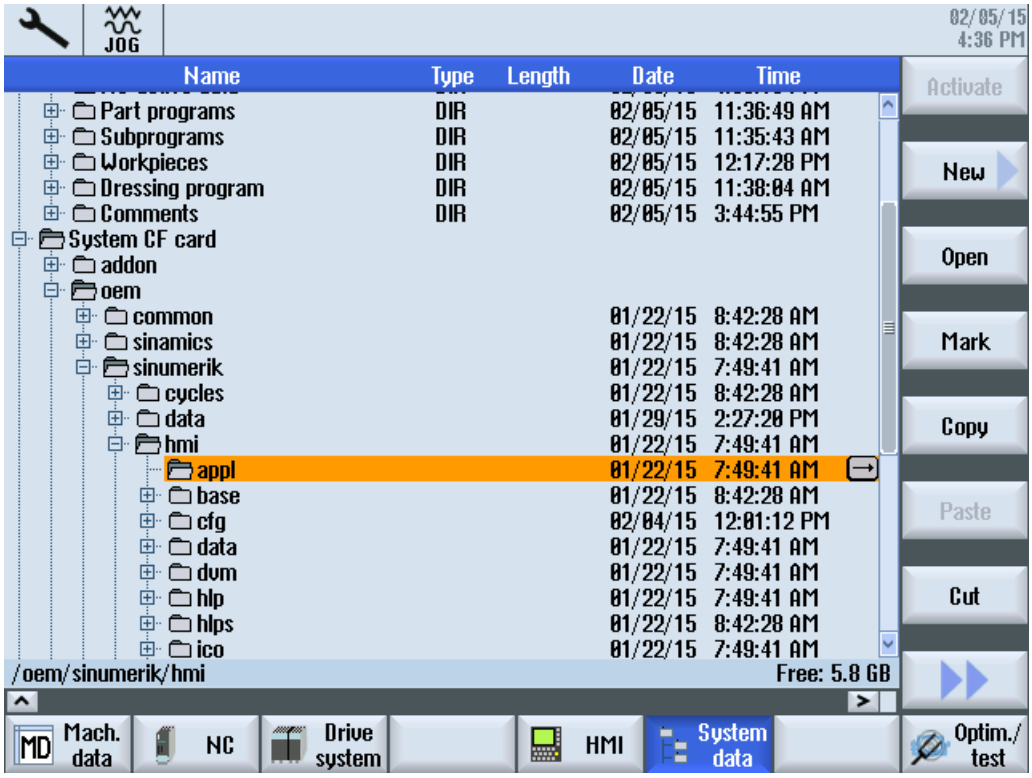



Figure 1-3 Manufacturer's folder

The following files in the manufacturer's folder "System CF-Card/oem/sinumerik/hmi" on the control are needed to configure user dialogs:

Table 1-2 Files for configuration

| File type | Name of the file | Meaning | Storage location in operating area Setup > System data |
|---|---|---|---|
| Script file | "xmldial.xml" | This script file uses XML tags to control how the configured softkey menus and dialog forms in SINUMERIK Operate will appear in the <CUSTOM> operating area. | Manufacturer's folder > sub-directory "appl" for the appli-cations |
| Text file | "oem_xml_screens_xxx.ts" | This text file contains the texts for the menus and dialog forms for the individual languages. | Manufacturer's folder > sub-directory "lng" for the lan-guages |

| File type | Name of the file | Meaning | Storage location in operating area Setup > System data |
|---|---|---|---|
| Bitmaps | | The control system supports BMP and PNG formats. | Manufacturer's folder > subdirectory "ico"<br>The bitmaps are saved in the subdirectory for the screen resolution belonging to the control.<br>Note: If a path to the bitmap file is specified, the files can be stored in this directory directly. |
| XML files inserted in the "xmldial.xml" control file with the "INCLUDE" XML tag. | E.g. "machine_settings.xml" | These files also contain programmed instructions for displaying the dialog forms and parameters in SINUMERIK Operate. | Manufacturer's folder > subdirectory "appl" for the applications |

## Dependencies of files for configuring user dialogs

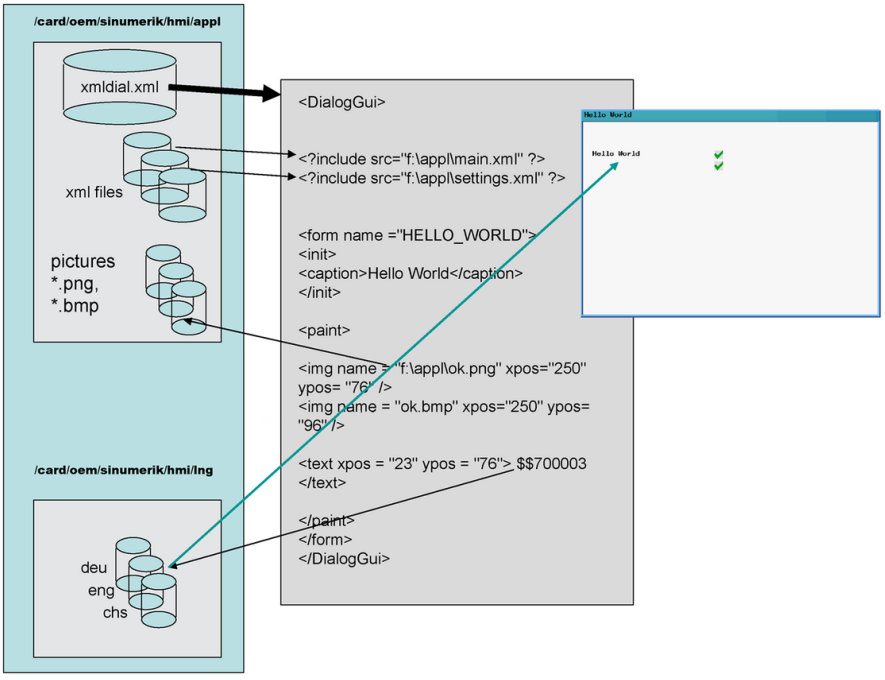



Figure 1-4 Dependencies

## Load configuration

As described in the "Storage location in the operating area" column in the previous "Files for configuration" table, the generated files must be copied to the appropriate subdirectories in the manufacturer's folder.

**Note**

As soon as there is a script file "xmldial.xml" in the subdirectory for applications, the user can start this user dialog in the <CUSTOM> operating area.

After the initial copying process, the control system needs to be reset via "Normal power-up".

## Example of a user dialog in SINUMERIK Operate

The configured softkey menus are displayed when the <CUSTOM> operating area is called. This enables the user to operate the dialog forms which have been configured.

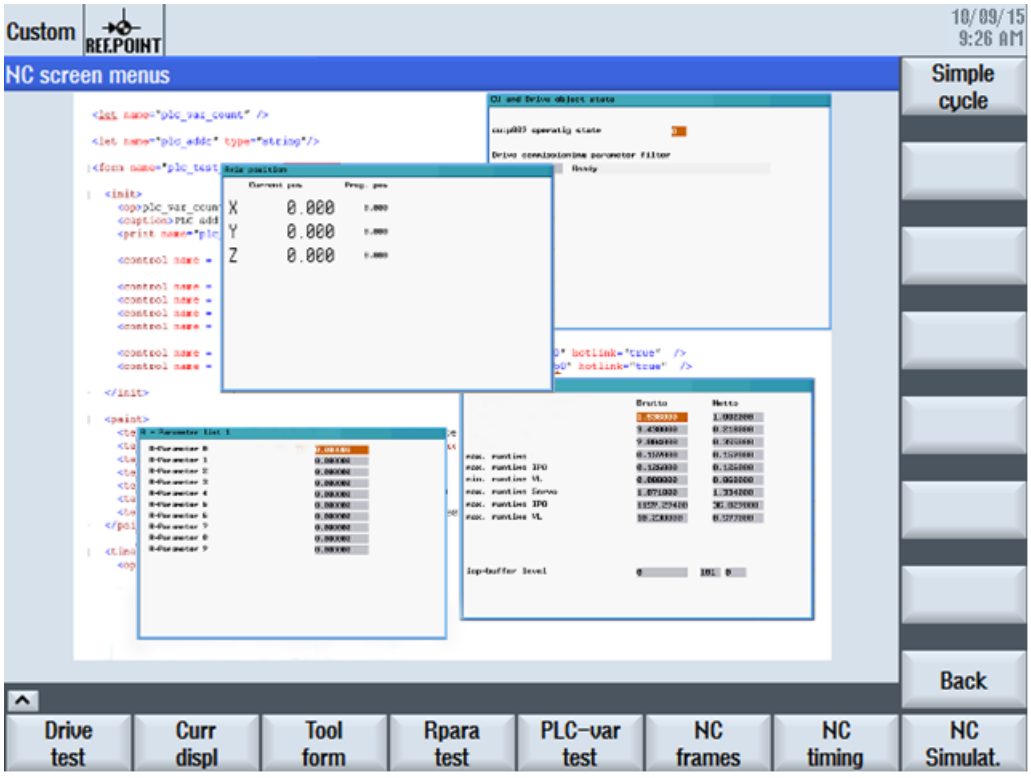



Figure 1-5 Example of a user dialog in the <CUSTOM> operating area

Further examples are provided in the toolbox.

## See also

Predefined functions (Page 107)

# 1.4 Structure of configuration file

## Overview

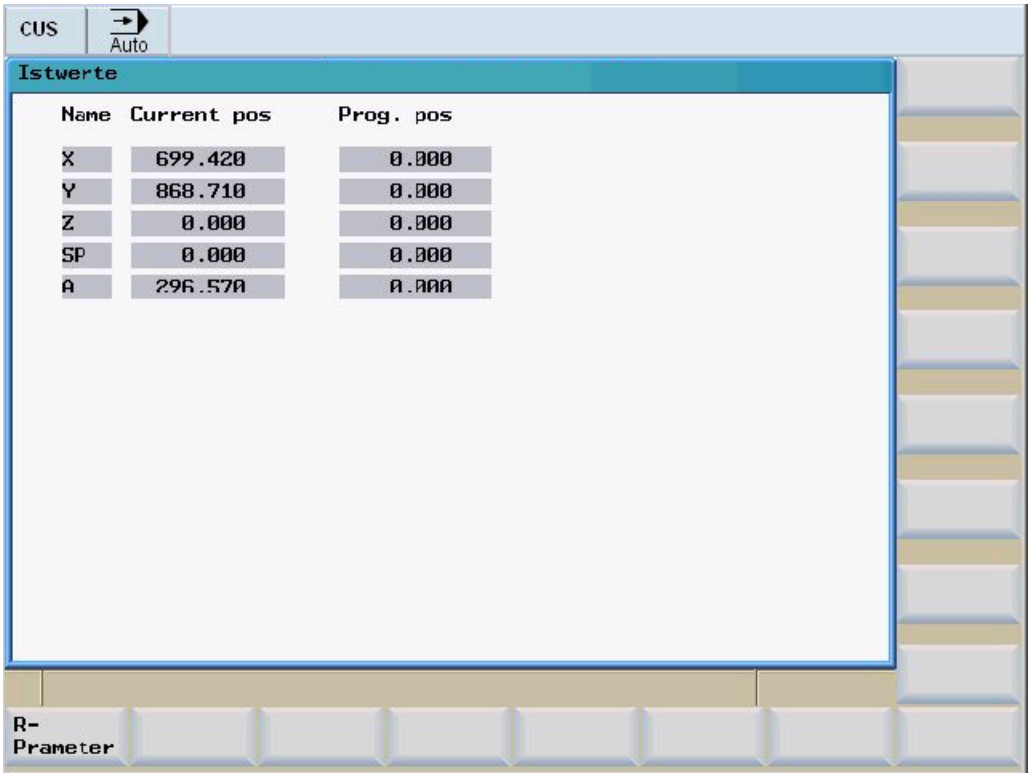
A configuration file consists of the following elements:

- Description of the "main" start menu with start softkeys
- Definition of the dialogs
- Definition of the variables
- Description of the blocks
- Definition of softkey bars

The following examples show the XML script for the "xmldial.xml" file and the corresponding screenshots.

The script contains the dialogs for displaying actual values and residual distances, as well as an R parameter list.

## Dialog form section "Actual values"



**xmldial.xml**

```
<DialogGui>

<!--
main menu
It is called by the system software. It starts the application.
The menu tag manages the soft key reactions. One input form can be assigned
to a menu tag.
-->
<menu name = "MAIN">
<OPEN_FORM name = "CURRENT_DISPLAY" />

<softkey POSITION="1">
<caption>R-%nPrameter</caption>
<navigation>MENU_R_PARAMETER</navigation> <!-- opens the menu R parameter
-->
</softkey>


</menu>


<form name = "CURRENT_DISPLAY">

<init>
<caption>Istwerte</caption>

<control name = "label1" xpos = "36" ypos = "56" width = "32"
fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[0]" />
<control name = "label2" xpos = "36" ypos = "76" width = "32"
fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[1]" />
<control name = "label3" xpos = "36" ypos = "96" width = "32"
fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[2]" />
<control name = "label4" xpos = "36" ypos = "116" width = "32"
fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[3]" />
<control name = "label5" xpos = "36" ypos = "136" width = "32"
fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[4]" />

<control name = "edit1" xpos = "80" ypos = "56" refvar="nck/Channel/
GeometricAxis/actProgPos[0]" hotlink="true" fieldtype="readonly"
format="%9.3f" time="super fast"/>
<control name = "edit2" xpos = "80" ypos = "76" refvar="nck/Channel/
GeometricAxis/actProgPos[1]" hotlink="true" fieldtype="readonly"
format="%9.3f" time="super fast"/>
<control name = "edit3" xpos = "80" ypos = "96" refvar="nck/Channel/
GeometricAxis/actProgPos[2]" hotlink="true" fieldtype="readonly"
format="%9.3f" time="super fast"/>
<control name = "edit4" xpos = "80" ypos = "116" refvar="nck/Channel/
GeometricAxis/actProgPos[3]" hotlink="true" fieldtype="readonly"
format="%9.3f" time="super fast"/>
<control name = "edit5" xpos = "80" ypos = "136" refvar="nck/Channel/
GeometricAxis/actProgPos[4]" hotlink="true" fieldtype="readonly"
format="%9.3f" time="super fast"/>
```

**xmldial.xml**

```
<control name = "edit11" xpos = "210" ypos = "56" refvar="nck/Channel/
GeometricAxis/progDistToGo[0]" hotlink="true" fieldtype="readonly"
format="%9.3f" time="super fast"/>
<control name = "edit12" xpos = "210" ypos = "76" refvar="nck/Channel/
GeometricAxis/progDistToGo[1]" hotlink="true" fieldtype="readonly"
format="%9.3f" time="super fast"/>
<control name = "edit13" xpos = "210" ypos = "96" refvar="nck/Channel/
GeometricAxis/progDistToGo[2]" hotlink="true" fieldtype="readonly"
format="%9.3f" time="super fast"/>
<control name = "edit14" xpos = "210" ypos = "116" refvar="nck/Channel/
GeometricAxis/progDistToGo[3]" hotlink="true" fieldtype="readonly"
format="%9.3f" time="super fast"/>
<control name = "edit15" xpos = "210" ypos = "136" refvar="nck/Channel/
GeometricAxis/progDistToGo[4]" hotlink="true" fieldtype="readonly"
format="%9.3f" time="super fast"/>

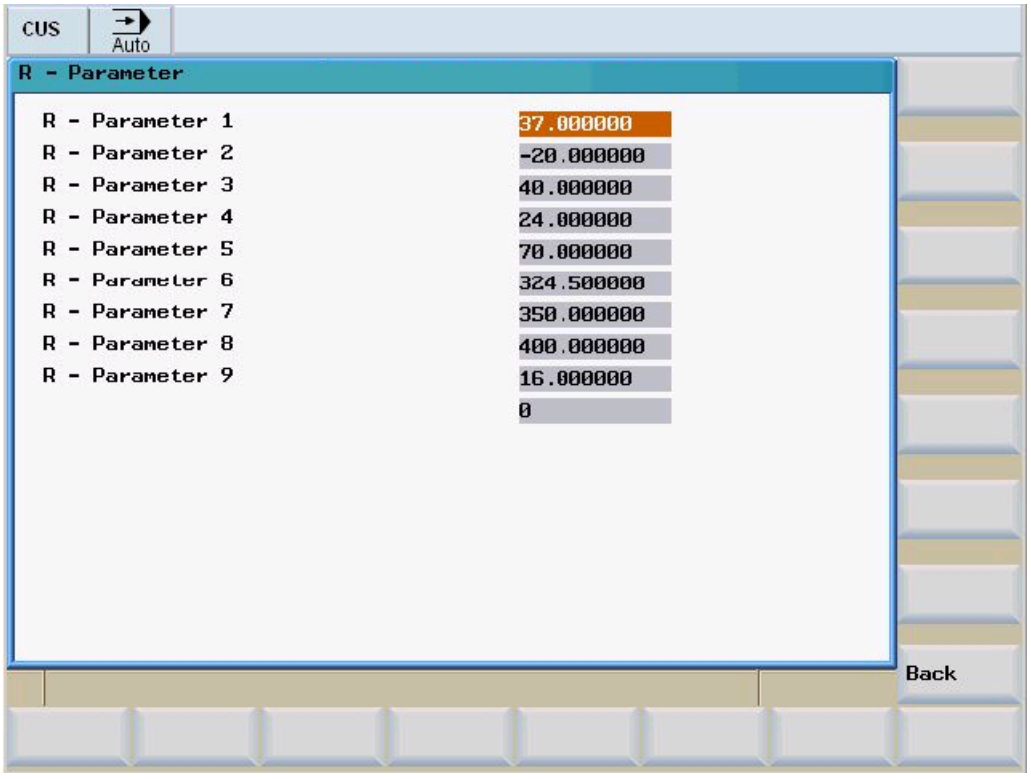
</init>

<paint>
<text xpos= "36" ypos="30">Name</text>
<text xpos= "80" ypos="30">Current pos</text>
<text xpos= "210" ypos="30">Prog. pos</text>

</paint>
</form>

.....
```

## Dialog form section "R Parameters"



**xmldial.xml**

```
.....

<!-- ************************************************************
Menu R- Parameter
-->

<menu name ="MENU_R_PARAMETER">
<OPEN_FORM name = "R-Parameter" />

<softkey POSITION="16">
<caption>Back</caption>
<navigation>MAIN</navigation>
</softkey>

</menu>

<form name = "R-Parameter">

<init>
<DATA_ACCESS type="true" />
<caption>R - Parameter</caption>

<control name = "edit1" xpos = "322" ypos = "34" refvar="nck/Channel/
Parameter/R[1]" />
<control name = "edit2" xpos = "322" ypos = "54" refvar="nck/Channel/
Parameter/R[2]" />
<control name = "edit3" xpos = "322" ypos = "74" refvar="nck/Channel/
Parameter/R[3]" />
<control name = "edit4" xpos = "322" ypos = "94" refvar="nck/Channel/
Parameter/R[4]" />
<control name = "edit5" xpos = "322" ypos = "114" refvar="nck/Channel/
Parameter/R[5]" />
<control name = "edit6" xpos = "322" ypos = "134" refvar="nck/Channel/
Parameter/R[6]" />
<control name = "edit7" xpos = "322" ypos = "154" refvar="nck/Channel/
Parameter/R[7]" />
<control name = "edit8" xpos = "322" ypos = "174" refvar="nck/Channel/
Parameter/R[8]" />
<control name = "edit9" xpos = "322" ypos = "194" refvar="nck/Channel/
Parameter/R[9]" />
<control name = "edit10" xpos = "322" ypos = "214" refvar="nck/Channel/
Parameter/R[10]" />
<control name = "edit10" xpos = "322" ypos = "214" refvar="plc/mb170" />
</init>

<paint>
<text xpos = "23" ypos = "34">R - Parameter 1</text>
<text xpos = "23" ypos = "54">R - Parameter 2</text>
<text xpos = "23" ypos = "74">R - Parameter 3</text>
<text xpos = "23" ypos = "94">R - Parameter 4</text>
<text xpos = "23" ypos = "114">R - Parameter 5</text>
<text xpos = "23" ypos = "134">R - Parameter 6</text>
<text xpos = "23" ypos = "154">R - Parameter 7</text>
<text xpos = "23" ypos = "174">R - Parameter 8</text>
<text xpos = "23" ypos = "194">R - Parameter 9</text>
```

**xmldial.xml**

```
</paint>
</form>

</DialogGui>
```

## 1.5 Language dependency

Language-dependent texts are used for:

- Softkey labels
- Headers
- Help texts
- Any other texts

The language-dependent texts are stored in text files.

**Note**

You will need to perform the following steps when using these text files:

- Make them available in the required languages.
- Transfer them into the relevant language directories of the control system.

# 1.6 XML diagnostics

The system provides the Easy XML diagnostic function for diagnosing and finding script errors.

The use of the diagnostic function checks the XML syntax and runs through all the scripts belonging to the project. To do this, functions, menus, forms and the existence/validity of variables are also checked. The errors found are listed.

The Easy XML diagnostics function can be activated either by an attribute in the **DialogGui** tag or by Display machine data.

## Activate easy XML diagnostics

**Example:**

```
<DialogGui diagnose="true" >
…
…
</DialogGui >
```

or

**Display machine data:**

| MD9113 $MM_EASY_XML_DIAGNOSE | Diagnostics and correction help support for Easy XML scripts |
|---|---|
| = 0 | No diagnostics active |
| = 1 | Syntax check active |

## Softkey function overview

| Dialog | Softkey | Function |
|---|---|---|
| Main menu "EasyXML diagnostics" | Start script | The loaded program is started directly. |
| | Start check | The syntax check is being started. All accrued messages can be seen. Renewed checking is available. |
| "Errors" and "Warnings" dialogs | Errors | The result are displayed, sorted according to errors and warnings. |
| | Warnings | |
| | Go to error | If errors or warnings are found, the softkey is displayed.<br>The softkey opens the marked XML file from the error list for editing. The cursor is automatically placed on the erroneous row. |
| | Check result | All accrued messages can be seen. |

| Dialog | Softkey | Function |
|---|---|---|
| "Document" dialog | Save | Changes in the XML file are being saved. |
| | Cancel | The XML file is closed unchanged. The check result is displayed again. |

# 1.7 XML identifier

## 1.7.1 General structure

### Structure and instructions of the script file for dialog configuration

All dialog configurations should be stored in the **DialogGui** tag.

```
<DialogGui>
…
</DialogGui>
```

**Example:**

```
<?xml version="1.0" encoding="utf-8"?>
<DialogGui>
…
<FORM name ="Hello_World">
<INIT>
<CAPTION>Hello World</CAPTION>
</INIT>
...
</FORM>

</DialogGui>
```

### Instructions

The language offers the following instructions for executing conditional instructions and loop controls:

- For loop
- WHILE loop
- Do while loop
- Conditional processing
- Switch and Case instructions
- Operator controls in a dialog form
- Softkey descriptions
- Define variables

A detailed description of the instructions can be obtained in section "Instruction/identifier descriptions (Page 24)".

## 1.7.2 Instruction/identifier descriptions

The following **XML tags** are defined for generating dialogs and menus, and for executing program sequences:

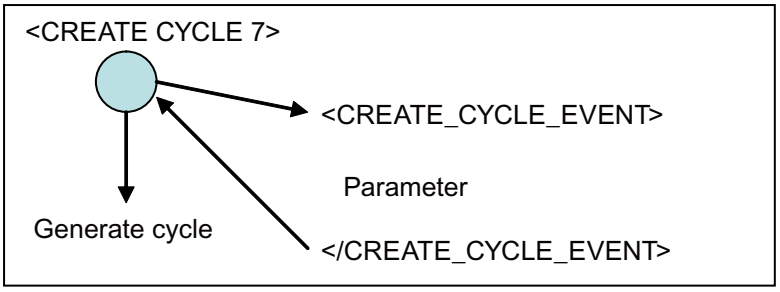
**Note**

Attribute values that are in quotation marks "<...>" should be replaced by the currently used expressions.

**Example:**
<DATA_LIST action="read/write/append" id="<list name>">
is programmed as follows:

<DATA_LIST action="read/write/append" id="my datalist">

| Tag identifier | Description |
|---|---|
| **BREAK** | Conditional cancellation of a loop. |
| **CONTROL** | The tag is used to generate control elements. The description is provided in Chapter "Generating softkey menus and dialog forms (Page 51)". |
| **CONTROL_RESET** | The tag enables one or more control components to be restarted.<br><br>**Syntax:**<br>`<CONTROL_RESET resetnc="TRUE" />`<br><br>**Attributes:**<br>• **RESETNC = "TRUE"**<br>The NC component is restarted<br>• **RESETDRIVE = "TRUE"**<br>The drive components are restarted. |

| Tag identifier | Description |
|---|---|
| **CREATE_CYCLE_EVENT** | If the parser starts to process the tag CREATE_CYCLE, initially, the message <CREATE_CYCLE_EVENT sent> is sent to the active form. This message can be used for preparing the cycle parameters, before the parser generates the NC operation from the parameter list and the generation rule.<br><br><br>**Syntax:**<br>`<CREATE_CYCLE_EVENT>`<br>…<br>…<br>`</CREATE_CYCLE_EVENT>`<br><br>**Example:**<br>`<SOFTKEY_OK>`<br>…<br>`<CREATE_CYCLE />`<br>…<br>…<br>`<SOFTKEY_OK>`<br>…<br>…<br><br>`<FORM>`<br>`<NC_INSTRUCTION>MY_CYCLE($P1, $P2)</ NC_INSTRUCTION >`<br>…<br>…<br>`<CREATE_CYCLE_EVENT>`<br>`<type_cast name="P1" type="int"/>`<br><br>`<OP> P1 = P1 * 150 </OP>`<br><br>`</CREATE_CYCLE_EVENT>`<br>…<br>…<br><br>`</FORM>` |

| Tag identifier | Description |
|---|---|
| **DATA** | The tag allows writing to NC, PLC, GUD and drive data.<br>Chapter "Component addressing (Page 94)" describes how addresses are formed.<br><br>**Attribute:**<br>• **name**<br>Variable address<br><br>**Tag value:**<br>All alphanumeric terms are approved as tag values. If a value is to be written directly from a local variable, the name of the local variable should be preceded by the $ replacement operator.<br>If the content of a local variable is to be written, the variable name should be preceded by the "$" replacement operator.<br><br>**Syntax:**<br>`<DATA name="<variable name>"> value </DATA>`<br><br>**Example:**<br>`<DATA name = "plc/mb170"> 1 </DATA>`<br>…<br>`<LET name = "tempVar"> 7 </LET>`<br>`<!– the contents of the local variables "tempVar" are written to bit memory byte 170 →`<br>`<DATA name = "plc/mb170">$tempVar</DATA>` |

| Tag identifier | Description |
|---|---|
| **DATA_LIST** | The tag enables the listed drive and machine data to be saved or restored.<br>Addresses are listed in lines. Chapter "Component addressing (Page 94)" describes how addresses are formed.<br>Up to 20 temporary data lists can be created.<br><br>**Attributes:**<br>• **action**<br>*read* – the values of the listed variables are stored in a temporary memory<br>*append* – the values of the listed variables are added to an existing list<br>*write* – the backed up values are copied to the relevant machine data<br>• **id**<br>The identifier identifies the temporary memory<br>**Syntax:**<br>`<DATA_LIST action="<read/write/append>"  id="<list name>">`<br>`NC/PLC address compilation`<br>`</DATA_LIST>`<br><br>**Example:**<br>`<DATA_LIST action ="read" id="<name>">`<br>`nck/channel/parameter/r[2]`<br>`nck/channel/parameter/r[3]`<br>`nck/channel/parameter/r[4]`<br>`$MN_USER_DATA_INT[0]`<br>…<br>`</ DATA_LIST>`<br>`<DATA_LIST action ="write" id="<name>" />` |
| **DO_WHILE** | Do while loop<br>`DO`<br>`  Instructions`<br>`WHILE (Test)`<br><br>**Syntax:**<br>`<DO_WHILE>`<br>`Instructions`<br>…<br>`<CONDITION>…</CONDITION>`<br>`</DO_WHILE>`<br>The Do While loop comprises a block of instructions and a condition. The code in the instruction block is executed first, then the condition is applied. If the condition is true, the function executes the code section again. This is continually repeated until the condition is false.<br><br>**Example:**<br>`<DO_WHILE>`<br>`<DATA name = "PLC/qb11"> 15 </DATA>`<br>`<CONDITIION> "plc/ib9" == 0 </CONDITION>`<br>`</DO_WHILE>` |

| Tag identifier | Description |
|---|---|
| **DYNAMIC_INCLUDE** | The tag includes an XML script file.<br>Contrary to the INCLUDE tag, read-in is first only realized when executing the corresponding operation.<br>For large projects, the use of the tag reduces the load time of the customer area and/or the cycle support. Further, the average level of resources is reduced, as not all of the dialogs are always called during a session.<br><br>**Syntax:**<br><DYNAMIC_INCLUDE src="path name"/><br><br>**Example:**<br><SOFTKEY POSITION="3"><br><CAPTION>MY_MENU</CAPTION><br><DYNAMIC_INCLUDE src/><br><NAVIGATION>MY_MENU</NAVIGATION><br></SOFTKEY> |
| **ELSE** | Instruction for situations where the condition has not been met (**IF, THEN, ELSE**) |

| Tag identifier | Description |
|---|---|
| **FOR** | For loop<br>`for (initialization; test; continuation) instruction(s)`<br><br>**Syntax:**<br>`<FOR>`<br>`<INIT>…</INIT>`<br>`<CONDITION>…</CONDITION>`<br>`<INCREMENT>…</INCREMENT>`<br>`Instructions`<br>`…`<br>`</FOR>`<br>The For loop is executed as follows:<br>1. Evaluation of the expression **initialization** (INIT).<br>2. Evaluation of the expression **test** (CONDITION) as a Boolean expression. If the value is false, the For loop is exited.<br>3. Execution of the following instructions.<br>4. Evaluation of the expression **continuation** (INCREMENT).<br>5. Continue with 2.<br>All the variables within the INIT, CONDITION, and INCREMENT branches are declared and initialized outside the For loop.<br><br>**Example:**<br>`<LET name = "count">0</LET>`<br>`<FOR>`<br>`<INIT>`<br>`<OP> count = 0</OP>`<br>`</INIT>`<br>`<CONDITION> count <= 7 </CONDITION>`<br>`<INCREMENT>`<br>`<OP> count = count + 1 </OP>`<br>`</INCREMENT>`<br>`<OP> "plc/qb10" = 1+ count </OP>`<br>`</FOR>` |
| **FORM** | The tag contains the description of a user dialog. The description is provided in Chapter "Generating softkey menus and dialog forms (Page 51)". |
| **HMI_RESET** | The tag triggers an HMI restart.<br>Interpretation is stopped after this instruction. |

| Tag identifier | Description |
|---|---|
| **IF** | Conditional instruction (**IF, THEN, ELSE**)<br>The THEN and ELSE tags are enclosed in the IF tag.<br>The condition that is executed in the CONDITION tag follows the IF tag. The further processing of the instructions depends upon the result of the operation. If the function result is true, then the THEN branch is executed and the ELSE branch is skipped. If the result of the function is false, the parser executes the ELSE branch.<br><br>**Syntax:**<br>`<IF>`<br>`<CONDITION> Condition != 7 </CONDITION>`<br>`<THEN>`<br>`Instruction for the case: Condition fulfilled`<br>`</THEN>`<br>`<ELSE>`<br>`Instruction for the case: Condition not fulfilled`<br>`</ELSE>`<br>`</IF>`<br><br>**Example:**<br>`<IF>`<br>`<CONDITION> "plc/mb170" != 7 </CONDITION>`<br>`<THEN>`<br>`<OP> "plc/mb170" = 7 </OP>`<br>…<br>`</THEN>`<br>`<ELSE>`<br>`<OP> "plc/mb170" = 2 </OP>`<br>…<br>`</ELSE>`<br>`</IF>` |
| **INCLUDE** | The instruction includes an XML description.<br>(see also DYNAMIC_INCLUDE in this table)<br><br>**Attribute:**<br>• **src**<br>Contains the path name.<br><br>**Syntax:**<br>`<?INCLUDE src="<Path name>" ?>` |

| Tag identifier | Description |
|---|---|
| **LET** | The instruction creates a local variable under the specified name.<br>**Fields:**<br>Using the attribute **dim** (dimension) single or two-dimensional fields can be created. The field index addresses the individual field elements.<br>For a two-dimensional field, initially the line index is specified and then the column index.<br>• One-dimensional field:<br>Indices 0 to 4<br>0<br>1<br>2<br>3<br>4<br>• Two-dimensional field:<br>Index line 0 to 3 and index column 0 to 5<br>0,0 0,1 0,2 0,3 0,4 0,5<br>1,0 1,1 1,2 1,3 1,4 1,5<br>2,0 2,1 2,2 2,3 2,4 2,5<br>3,0 3,1 3,2 3,3 3,4 3,5<br>**Attributes:**<br>• **name**<br>Variable name<br>• **type**<br>The variable type can be an integer (INT), unsigned integer (UINT), double (DOUBLE), float (FLOAT), string (STRING), or STRUCT. It is also possible to use a structure created by **typedef** as a variable type (see tag identifier TYPEDEF). If there is no type instruction specified, the system creates an integer variable.<br>`<LET name = "VAR1" type = "INT" />`<br>• **permanent**<br>If the attribute is set to **true**, then the variable value is saved permanently. This attribute is only effective for a global variable.<br>• **dim**<br>The following number of field elements must be specified. For a two-dimensional field, the second dimension is specified after the first dimension separated by a comma.<br>A field element is accessed via the field index, which is specified in square brackets after the variable name.<br>name[index] or name[row,column]<br>– One-dimensional field: dim="<Number of elements>"<br>– Two-dimensional field: dim="<Number of lines>,<number of columns>"<br>Non-initialized field elements are pre-assigned with "0". |

| Tag identifier | Description |
|---|---|
| **LET** Continued | **Example:**<br>One-dimensional field:<br>`<let name="array" dim="10"></let>`<br>Two-dimensional field:<br>`<let name="list_string" dim="10,3" type="string"></let>`<br><br>**Pre-assignment:**<br>A variable can be initialized with a value.<br>`<LET name = "VAR1" type = "INT"> 10 </LET>`<br><br>If values comprising NC or PLC variables are saved in a local variable, the assignment operation automatically adapts the format to that of the variables which have been loaded.<br>• Pre-assignment for a string variable:<br>Texts containing more than one line can be assigned to a string variable if the formatted text is transferred as a value. If a line is to end with a line feed <LF> (line feed) , the characters "\\n" should be added at the end of the line.<br>`<LET name = "text" type = "string"> F4000 G94\\n`<br>`G1 X20\\n`<br>`Z50\\n`<br>`M2\\n`<br>`</LET>>`<br><br>**`Fields (Arrays):`**<br>`<let name="list" dim="10,3">`<br>`{1,2,3},`<br>`{1,20}`<br>`</let>`<br><br>`<let name="list_string" dim="10,3" type="string">`<br>`{"text 10","text 11"},`<br>`{"text 20","text 21"}`<br>`</let>`<br><br>**Assignment:**<br>Values made up of the machine data or subroutines can be assigned to a variable using the assignment operation "=".<br>A variable remains valid until the end of the higher-level XML block.<br>Variables which are to be available globally should be created directly after the **DialogGUi** tag.<br>The following must be observed for a dialog box:<br>• The message processing opens the corresponding tag.<br>• The tag is closed after the message has been executed.<br>• All variables within the tag are deleted when closing. |

| Tag identifier | Description |
|---|---|
| **LET** Continued | **Variable type struct**:<br>This variable type contains a composition of variables that can be addressed using the structure name. A structure can contain all variable types and structures.<br>Within the structure, a variable is declared with the "element" tag. The attributes of the tags and the initialization correspond to the attributes and initialization of the let instruction.<br><br>`<let name="name" type="struct">`<br>`<element name="<Name>" type="<Variable type>" />`<br>…<br>…<br>…<br><br>`</let>` |
| **LET** Continued | Access to a variable of the structure is via the structure name and variable name. Both names are separated by a point operator.<br><br>`<op>`<br>`Structure_name.variable_name = value;`<br>`</op>`<br><br>**Example:**<br>`<let name="info" type="struct">`<br>`  <element name="id" type="int" />`<br>`  <element name="name" type="string" />`<br>`  <element name="phone" type="string" />`<br>`</let>`<br><br>`<op>`<br>`  info.id = 1;`<br>`  info.name = _T"my name";`<br>`  info.phone = _T"0034 45634";`<br>`</op>`<br><br>**Initialization of structures:**<br>Structures can be initialized when the variables are created by specifying an initial value for each structure element. In an array of structures, each structure must be separated from the others by braces. |

| Tag identifier | Description |
|---|---|
| **LOCK_OPERATING_AREA** | The operating area switchover is locked.<br>The operating area switchover lock is withdrawn with the **UNLOCK_OPERATING_AREA** tag.<br><br>**Syntax:**<br>`<LOCK_OPERATING_AREA />` |
| **MSG** | The operator component shows the message which is indicated in the tag.<br>If an alarm number is used, the dialog box displays the text which is saved for the number.<br><br>**Example:**<br>`<MSG text ="my message" />` |
| **MSGBOX** | The instruction opens a message box whose return value can be used for branching.<br><br>**Syntax:**<br>`<MSGBOX text="<Message>" caption="<caption>" retvalue="<variable name>" type="<button type>" />`<br><br>**Attributes:**<br>• **text**<br>Text<br>• **caption**<br>Header<br>• **retvalue**<br>Name of the variables to which the return value is copied:<br>1 – OK<br>0 – CANCEL<br>• **type**<br>Acknowledgment options:<br>"BTN_OK"<br>"BTN_CANCEL"<br>"BTN_OKCANCEL"<br>If an alarm number is used for the **"text"** or **"caption"** attribute, the message box displays the text which is saved for the number.<br><br>**Example:**<br>`<MSGBOX text="Test message" caption="Information" retvalue="result" type="BTN_OK" />` |

| Tag identifier | Description |
|---|---|
| **OP** | The tag executes the specified operations.<br>The operations listed in Chapter "Operators (Page 49)" can be executed.<br>For the purpose of accessing the NC, PLC, and drive data, the complete variable name should be placed in **quotation marks**. The "Component addressing" (Page 94) section contains details on address formation.<br>`PLC: "PLC/MB170"`<br>`NC: "NC/Channel/…"`<br><br>**Example:**<br>`<LET name = "tmpVar" type="INT"> </LET>`<br>`<OP> tmpVar = "plc/mb170" </OP>`<br>`<OP> tmpVar = tmpVar *2 </OP>`<br>`<OP> "plc/mb170" = tmpVar </OP>`<br><br>More than one equation can be used within an operation tag. A semicolon marks the end of the instruction.<br><br>**Example:**<br>`<op>`<br>`  x = x+1;`<br>`  y = y+1;`<br>`</op>`<br><br>**Character string processing:**<br>The operation instruction is able to process character strings and assign the results to the string variable specified in the equation.<br>The identifier _T should be placed at the start as a means of identifying text terms. Formatting of variable values is also possible. The identifier _F should be placed at the start of the formatting regulation, followed by the format instruction. The address is then specified for the variable.<br><br>**Example:**<br>`<LET name="buffer" type="string"></LET>`<br>…<br>…<br>…<br>`<OP> buffer = _T"unformatted value R0= " + "nck/Channel/Parameter/R[0]" + _T" and " + _T"$$85051" + _T" formatted value R1 " + _F %9.3f"nck/Channel/Parameter/R[1]" </OP>` |

| Tag identifier | Description |
|---|---|
| OPERATION | **Operation**<br>Instructions can be moved within an equation.<br>**Move left "<<" operator**<br>The << function moves bits to the left. You can specify the value and the number of move increments directly or with a variable. If the limit of the data format is reached, the bits will be moved beyond the limit without an error message being issued.<br>**Execution rule**<br>Execution from left to right; '+' and '-' have priority over '<<' and '>>'<br><br>**Example:**<br>`<op> idx = idx << 2 </op>`<br>`<op> idx = 3 + idx << 2 </op>`<br><br>**Move right ">>" operator**<br>The >> function moves bits to the right. You can specify the value and the number of move increments directly or with a variable. If the limit of the data format is reached, the bits will be moved beyond the limit without an error message being issued.<br>**Execution rule**<br>Execution from left to right; '+' and '-' have priority over '<<' and '>>'<br><br>**Example:**<br>`<op> idx = idx >> 2 </op>`<br>`<op> idx = 3+idx >> 2</op>` |
| PASSWORD | This tag is used to unlock additional units for the "EasyExtend" function.<br>The specified character string is written to the specified reference variable and must be processed in the PLC user program.<br><br>**Syntax:**<br>`<PASSWORD refVar ="<variable name>" />`<br><br>**Attributes:**<br>• **refVar**<br>Name of the reference variable<br>• **text**<br>Specifying an attribute replaces the default text (optional)<br><br>**Example:**<br>`<PASSWORD refvar="plc/mw107" />`<br><br>**Example optional:**<br>`<password refVar = "plc/MD108" text="Password" />` |
| POWER_OFF | A message prompts the operator to switch the machine off. The message text is permanently saved in the system. |

| Tag identifier | Description |
|---|---|
| **PRINT** | The tag outputs a text in the dialog line or copies the text to the variable specified.<br>If the text contains formatting identifiers, the variable values are inserted at the appropriate places.<br><br>**Syntax:**<br>`<PRINT name="Variable name" text="text %Formatting"> Variable, … </PRINT>`<br>`<PRINT text="text %Formatting"> Variable, … </PRINT>`<br>**Attributes:**<br>• **name**<br>Name of the variable where the text is to be stored (optional)<br>• **text**<br>Text<br><br>**Formatting:**<br>The character "%" causes the variable specified as the value to be formatted.<br>%[Flags] [Width] [.decimal places] type<br>• Flags:<br>Optional character for defining output formatting:<br>– Right-justified or left-justified ("-" for left-justified)<br>– Add leading zeros ("0")<br>– Fill with blanks<br>• Width:<br>The argument defines the minimum output width for a non-negative number. If the value to be output has fewer places than the argument defined, the missing spaces are filled with blanks.<br>• Decimal places:<br>With floating-point numbers, the optional parameter defines the number of decimal places.<br>• Type:<br>The type character defines which data formats are transferred for the print instruction. These characters need to be specified.<br>– d: Integer value<br>– f: Floating-point number<br>– s: String |

| Tag identifier | Description |
|---|---|
| **PRINT** Continued | **Values:**<br>Number of variables whose values are to be inserted into the text.<br>The variable types must match the corresponding type identifier for the formatting instruction and must be separated from one another with a comma.<br><br>**Example:**<br>Output of a text in the information line<br>`<PRINT text="Infotext" />`<br>Output of a text with variable formatting<br>`<LET name="trun_dir"></LET>`<br>`<PRINT text="M%d">trun_dir</PRINT>`<br>Output of a text in a string variable with variable formatting<br>`<LET name="trun_dir"></LET>`<br>`<LET name="str" type="string" ></LET>`<br>`<print name="str" text="M%d ">trun_dir</print>` |
| **PROGRESS_BAR** | The tag opens or closes a progress bar. The bar is displayed below the application window.<br><br>**Syntax:**<br>`<PROGRESS_BAR type="<true/false>"> value </ PROGRESS_BAR>`<br><br>**Attributes:**<br>• **type =** "TRUE" - opens the progress bar<br>• **type** = "FALSE" - closes the progress bar<br>• **min**<br>(optional) – minimum value<br>• **max**<br>(optional) – maximum value<br><br>**Value:**<br>• **Value**<br>Percentage position of the bar<br>**Example:**<br>`<PROGRESS_BAR type="true" min="0" max="101" >20< /`<br>`PROGRESS_BAR>......<PROGRESS_BAR >50< /`<br>`PROGRESS_BAR>......<PROGRESS_BAR type="false" >100< /PROGRESS_BAR>` |

| Tag identifier | Description |
|---|---|
| **SEND_MESSAGE** | The tag sends a message with two parameters to the active form, which is processed in the tag message.<br><br>**Syntax:**<br>`<SEND_MESSAGE>p1, p2</SEND_MESSAGE>`<br><br>**Example:**<br>`<SOFTKEY POSITION="3">`<br>`  <caption>Set%nParameter</caption>`<br>`  <send_message>1, 0</send_message>`<br>`</SOFTKEY>`<br><br>`<FORM>`<br>…<br>…<br>`<MESSAGE>`<br>`<SWITCH>`<br>`<CONDITION>$message_par1</CONDITION>`<br>`<CASE value="1">`<br>…<br>…<br>`</CASE>`<br>`</SWITCH>`<br>`</MESSAGE>`<br>…<br>…<br><br>`</FORM>` |
| **SHOW_CONTROL** | The visibility of a control can be controlled using the tag.<br><br>**Syntax:**<br>`<SHOW_CONTROL name="<name>" type="<type>" />`<br><br>**Attributes:**<br>• **name**<br>Name of the control<br>• **type =** "TRUE" - control becomes visible<br>• **type** = "FALSE" - control becomes invisible (hidden)<br><br>**Example:**<br>`<SHOW_CONTROL name="myEditfield" type="false" />`<br>…<br>…<br>`<SHOW_CONTROL name="myEditfield" type="true" />` |

| Tag identifier | Description |
|---|---|
| **SLEEP** | The tag interrupts script execution for the specified period. The interruption time is obtained from the transferred value multiplied by the time base of 50 ms.<br><br>**Syntax:**<br>`<SLEEP value="Interruption time" />`<br><br>**Example:**<br>Wait time, 1.5 sec.<br>`<SLEEP value="30" />` |
| **STOP** | Interpretation is canceled at this point. |
| **SWITCH** | The **SWITCH** instruction describes a multiple choice. A term is evaluated once and compared with a number of constants. If the expression matches the constants, the instructions are executed within the **CASE** instruction.<br>The **DEFAULT** instruction is executed when none of the constants match the expression.<br><br>**Syntax:**<br>`<SWITCH>`<br>`<CONDITION> Value </CONDITION>`<br>`<CASE value="<Constant 1>">`<br>`Instructions`<br>`…`<br>`</CASE>`<br>`<CASE value="<Constant 2>">`<br>`Instructions`<br>`…`<br>`</CASE>`<br>`<DEFAULT>`<br>`Instructions`<br>`…`<br>`</DEFAULT>`<br>`</SWITCH>` |

| Tag identifier | Description |
|---|---|
| **SWITCHTOAREA** | The SWITCHTOAREA tag changes from the Customer area into the specified operating area.<br>The parameter is specified as an attribute value.<br><br>**Syntax:**<br>`<switchToArea name="area" args="argument "/>`<br><br>**Attributes:**<br>**name**<br>The following names are declared for the operating areas:<br>• **AreaMachine** - Machine<br>• **AreaParameter** - Parameters<br>• **AreaProgramEdit** - Editor<br>• **AreaProgramManager** - Program Manager<br>• **AreaDiagnosis** - Diagnostics<br>• **AreaStartup** - Setup<br>**args** (reserved)<br>Runtime arguments can be passed to the operating area to activate dialogs.<br><br>**Example:**<br>`<switchToArea name="AreaMachine" />` |
| **SWITCHTODYNAMICTAR-GET** | If the previous dialog is defined as a dynamic jump destination, the SWITCHTODYNAMIC-TARGET tag will activate this dialog and end script execution.<br><br>**Syntax:**<br>`<switchToDynamicTarget />` |
| **THEN** | Instruction if the condition has been fulfilled (**IF, THEN, ELSE**) |
| **TYPE_CAST** | The tag converts the data type of a local variable.<br><br>**Syntax:**<br><type_cast name="variable name" type=" new type" /><br>**Attributes:**<br>• **name**<br>Variable name<br>• **type**<br>The new data type is assigned to the variable.<br>• **convert**<br>The new data type is assigned to the variable. The variable value is also converted to the new data type. |

| Tag identifier | Description |
|---|---|
| **TYPEDEF** | A new identifier for a data type can be defined with this tag. This has the benefit for the structure definitions that the data type can be defined once and then used as a data type in a LET instruction.<br>The identifier and type are expected as attributes.<br>The parser supports only the specification of structure definitions.<br><br>In the type definition, a variable is declared with the "element" tag. The attributes of the tag correspond to the attributes of the let instruction.<br><br>`<typedef name="<identifier>" type="struct">`<br>`<element name="<name>" type="<variable type>" />`<br>…<br>…<br>…<br>`</typedef>`<br><br>After definition, the identifier can be used as a data type for the LET instruction.<br><br>`<let name="<variable name>" type="<identifier>"></let>`<br><br>**Example**:<br>`<typedef name="my_struct" type="struct">`<br>`<element name="id" type="int" />`<br>`<element name="name" type="string" />`<br>`<element name="phone" type="string" />`<br>`</typedef>`<br><br><br>`<let name="info" type="my_struct"></let>`<br>…<br>…<br>`<op>`<br>`info.id = 1;`<br>`info.name = _T"my name";`<br>`info.phone= _T"0034 45634";`<br>`</op>` |

| Tag identifier | Description |
|---|---|
| **TYPEDEF** Continued | Some predefined functions expect variables of structure type RECT, POINT, or SIZE as the call parameter. These structures are defined in the file struct_def.xml.<br><br>**RECT:**<br>`<typedef name="StructRect" type="struct" >`<br>`  <element name="left" type="int">0</element>`<br>`  <element name="top" type="int">0</element>`<br>`  <element name="right" type="int">0</element>`<br>`  <element name="bottom" type="int">0</element>`<br>`</typedef>`<br><br>**POINT:**<br>`<typedef name="StructPoint" type="struct" >`<br>`  <element name="x" type="int">0</element>`<br>`  <element name="y" type="int">0</element>`<br>`</typedef>`<br><br>**SIZE:**<br>`<typedef name="StructSize" type="struct" >`<br>`  <element name="width" type="int">0</element>`<br>`  <element name="height" type="int">0</element>`<br>`</typedef>` |
| **UNLOCK_OPERAT-ING_AREA** | Withdrawal of operating area switchover lock<br><br>**Syntax:**<br>`<UNLOCK_OPERATING_AREA />` |
| **WAITING** | The tag waits for the component to undergo a hot restart after an NC or drive reset.<br><br>**Attributes:**<br>• **WAITINGFORNC** = "TRUE" - the system waits for the NC to restart<br>• **WAITINGFORDRIVE** = "TRUE" - the system waits for the drives to restart<br><br>**Syntax:**<br>`<WAITING WAITINGFORNC = "TRUE"/>`<br><br>**Example:**<br>...<br>`<CONTROL_RESET resetnc = "true" resetdrive = "true"/>`<br>`<WAITING waitingfornc = "true" waitingfordrive = "true" />`<br>... |

| Tag identifier | Description |
|---|---|
| **WHILE** | WHILE loop<br>`WHILE (Test)`<br>`  Instruction`<br><br>**Syntax:**<br>`<WHILE>`<br>`<CONDITION>…</CONDITION>`<br>`Instructions`<br>`…`<br>`</WHILE>`<br>The While loop executes a sequence of instructions repeatedly while a condition is met. This condition is tested before the sequence of instructions is executed.<br><br>**Example:**<br>`<WHILE>`<br>`<CONDITION> "plc/ib9" == 0 </CONDITION>`<br>`<DATA name = "PLC/qb11"> 15 </DATA>`<br>`</WHILE>` |
| **XML_PARSER** | The "XML_PARSER" tag can be used to parse XML files.<br>The parser interprets an XML file and calls defined call-back functions. Each call-back function belongs to a predefined event. The programmer can process the XML data within this function.<br><br>**Predefined events:**<br>• start document<br>The parser opens the document and starts parsing.<br>• end document<br>The parser closes the document.<br>• start element<br>The parser has found an element and creates a list with all attributes and attribute values. These lists are forwarded to the call-back function.<br>• end element<br>The end of the element has been found.<br>• characters<br>The parser forwards all characters of an element.<br>• error<br>The parser has detected a syntax error.<br><br>When an event occurs, the parser calls the callback function and checks the return value of the function. If the function returns the value "true", the parser continues the process. |

| Tag identifier | Description |
|---|---|
| **XML_PARSER** Continued | **Interfaces**<br>The value of the **name** attribute contains the path to the XML file.<br>To assign events to the call-back functions, the following properties must be specified:<br>**Standard**<br>startElementHandler<br>endElementHandler<br>charactersHandler<br>**Optional**<br>errorHandler<br>documentHandler<br>The value of an attribute defines the name of the call-back function.<br><br>**Example:**<br>`<XML_PARSER name="f:\appl\xml_test.xml">`<br>`<!-- standard handler -->`<br>`<property startElementHandler="startElementHandler" />`<br>`<property endElementHandler="endElementHandler" />`<br>`<property charactersHandler="charactersHandler" />`<br>`<!-- optional handler -->`<br>`<property errorHandler="errorHandler" />`<br>`<property documentHandler="documentHandler" />`<br>`</XML_PARSER>` |

| Tag identifier | Description |
|---|---|
| **XML_PARSER** Continued | The parser also supplies variables so that the call-back functions can access the event data.<br><br>**startElementHandler:**<br>**Function parameters**<br>tag_name - tag name<br>num - number of attributes found<br>**System variables**<br>$xmlAttribute String array with the number of elements indicated by num.<br>$xmlValue String array that contains the 0-num attribute value range.<br><br>**Example:**<br>`<function_body name="startElementHandler" return="true" parameter="tag_name, num">`<br>`  <print text="attribute name %s"> $xmlAttribute[0]</print>`<br>`  <print text="attribute value %s"> $xmlValue[0]</print>`<br>`</function_body>`<br><br>**endElementHandler:**<br>**Function parameters**<br>tag_name - tag name<br><br>**Example:**<br>`<function_body name="endElementHandler" parameter="tag_name">`<br>`  <print text="name %s"> tag_name </print>`<br>`</function_body>` |

| Tag identifier | Description |
|---|---|
| **XML_PARSER** Continued | **charactersHandler:**<br>**System variables**<br>$xmlCharacters String with data<br>$xmlCharactersStart Always 0<br>$xmlCharactersLength Number of bytes<br><br>**Example:**<br>`<function_body name="charactersHandler" return="true" >`<br>`<print text="chars %s"> $xmlCharacters </print>`<br>`</function_body>`<br><br>**documentHandler:**<br>**Function parameters**<br>state 1 start document, 2 end document<br><br>**errorHandler:**<br>**System variables**<br>$xmlErrorString Contains the invalid line (system variable)<br><br>**Example:**<br>`<function_body name="errorHandler" >`<br>`<print text="error %s">$xmlErrorString</print>`<br>`</function_body>`<br><br>**Call-back result:**<br>$return If 1 (true), the parser continues parsing the file |

## 1.7.3 Color coding

The color attribute uses the color coding scheme for the HTML language.

In terms of syntax, color specifications consist of the "#" (hash) character and six digits from the hexadecimal system, with each color represented by two digits.

R – Red

G – Green

B – Blue

**#RRGGBB**

**Example:**

color= "#ff0011"

**Example colors:**

| Red | Green | Blue | Yellow | White | Black |
|---|---|---|---|---|---|
| #FF0000 | #00FF00 | #0000FF | #ffff00 | #FFFFFF | #000000 |

## 1.7.4 Special XML syntax

Characters with special meanings in XML syntax have to be rewritten if they are to be displayed correctly by a general XML editor.

The following characters are affected:

| Character | Notation in XML |
|---|---|
| < | < |
| > | > |
| & | & |
| " | " |
| ' | ' |

### 1.7.5 Operators

The operation instruction processes the following operators:

| Operator | Meaning |
|---|---|
| = | Assignment |
| == | Equal to |
| <, < | Less than |
| >, > | Greater than |
| <=, <= | Less than or equal to |
| >=, >= | Greater than or equal to |
| \| | OR operation in bits |
| \|\| | Logic OR operation |
| &, & | AND operation in bits |
| &&, && | Logic AND operation |
| + | Addition |
| - | Subtraction |
| * | Multiplication |
| / | Division |
| ! | Not |
| != | Not equal to |

Operation instructions are processed from left to right. It may make sense to place terms in parentheses under certain circumstances in order to define the priority for executing subterms.

### 1.7.6 System variables

System variables are variables available in every script, which are used for exchanging data between the parser and the script execution.

The following table provides an overview of the tags for which variables are automatically generated.

| Variable name | Meaning | Valid in tag |
|---|---|---|
| $actionresult | Signals to the parser, whether the parser is to process the event. | KEY_EVENT |
| $focus_name | Contains the name of the field which has the input focus | FOCUS_IN<br>INDEX_CHANGED<br>EDIT_CHANGED |
| $focus_item_data | Contains the numerical value item_data, that was assigned to the field | |
| $gestureinfo | For finger gestures, the parser provides the gesture information in the variable and executes the gesture_event tag. | GESTURE_EVENT |
| $return | The variable transports the return value of a subfunction | FUNCTION_BODY |
| $message_par1<br>$message_par2 | Contains the call parameters of the SEND_MESSAGE function | MESSAGE |
| $xmlAttribute | Contains a list of the tag attributes found | XML_PARSER<br>startElementHandler |
| $xmlValue | Contains a list of the tag values found | |
| $xmlCharacters | Contains the data flow | XML_PARSER<br>charactersHandler |
| $xmlCharactersStart | Contains the start index: | |
| $xmlCharactersLength | Contains the number of characters stored in the data flow | |
| $mouse_event.type<br>$mouse_event.x<br>$mouse_event.y<br>$mouse_event.id<br>$mouse_event.buttons<br>$mouse_event.button | Structure to transfer the parameters of the mouse event | MOUSE_EVENT |

## 1.7.7 Generating softkey menus and dialog forms

User menus can only be inserted if there is a main-menu tag with the name "main" in the XML description. This tag is called by the system after the <CUSTOM> operating area has been activated. Further menu branches and dialog-box activation can be defined within the tag.

```
<menu name= "MAIN">
<OPEN_FORM name= "main dialogue">
<softkey POSITION="1">
<caption>sub menu 1</caption>
<navigation>sub menu 1</navigation>
</softkey>
<softkey POSITION="8">
<caption>sub menu 8</caption>
<navigation>sub menu 8</navigation>
</softkey>
</menu>

 <menu name= "sub menu 1">
 <OPEN_FORM name= "dialogue 1">
 </menu>

<menu name= "sub menu 8">
<OPEN_FORM name= "dialogue 8“>
</menu>
```

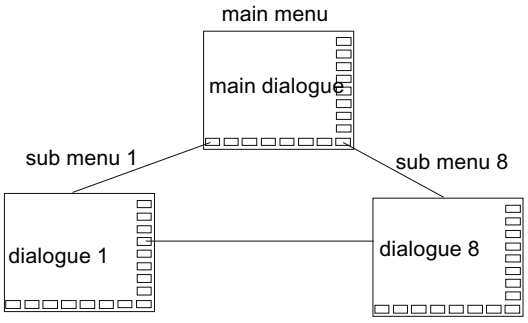
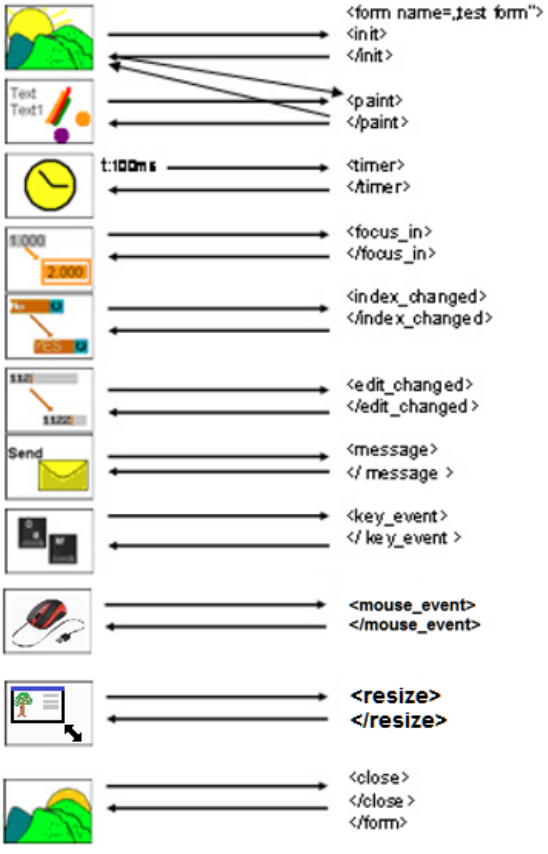



Figure 1-6 Menu structure

| Tag identifier | Description |
|---|---|
| **FORM** | The tag contains the description of a user dialog. The relevant tags are described in the section on generating menus and dialog forms.<br><br>**Syntax:**<br>`<FORM name="<dialog name>" color="#ff0000">`<br><br>**Attributes:**<br>• **color**<br>Background color of the dialog form (color coding, see Chapter "Color coding (Page 48)")<br>– Default white<br>• **name**<br>Identifier of the form<br>• **type**<br>Permissible value is *cycle* , which identifies a user cycle screen form<br>• **xpos**<br>X-position of the top left corner of the dialog box (optional)<br>• **ypos**<br>Y position of the top left corner (optional)<br>• **width**<br>Extension in the X direction (in pixels) (optional)<br>• **height**<br>Extension in the Y direction (in pixels) (optional) |
| **FORM** Continued | With the AUTOSCALE_CONTENT attribute, you can define how the controls of a form will behave at different screen resolutions. By default, the controls are automatically adapted to the screen resolution already set. If you would like to control the positioning and dimensioning yourself, set value OFF for the attribute in the Form tag.<br><br>**Attribute:**<br>**autoscale_content**<br><br>**Values:**<br>• **on**<br>The coordinates of the controls are automatically adapted to the screen resolution (default)<br>• **off**<br>The coordinates of the controls are used unchanged<br><br>**Example:**<br>`<form name="main_form" autoscale_content="off">`<br>…<br>…<br>…<br>`</form>` |

| Tag identifier | Description |
|---|---|
| **FORM** continued | **Dialog messages:**<br>• **INIT**<br>• **PAINT**<br>• **TIMER**<br>• **CLOSE**<br>• **FOCUS_IN**<br>• **INDEX_CHANGED**<br>• **EDIT_CHANGED**<br>• **GESTURE_EVENT**<br>• **KEY_EVENT**<br>• **MESSAGE**<br>• **MOUSE_EVENT**<br>• **RESIZE** |
| **FORM** continued |  |

| Tag identifier | Description |
|---|---|
| **FORM** continued | **Syntax:**<br>`<FORM name = "<dialog name>" color = "#ff0000">`<br>**Example:**<br>`<FORM name = "R-Parameter">`<br>`<INIT>`<br>`<DATA_ACCESS type = "true" />`<br>`<CAPTION>R - Parameter</CAPTION>`<br>`<CONTROL name = "edit1" xpos = "322" ypos = "34" refvar = "nck/Channel/Parameter/R[1]" />`<br>`<CONTROL name = "edit2" xpos = "322" ypos = "54" refvar = "nck/Channel/Parameter/R[2]" />`<br>`<CONTROL name = "edit3" xpos = "322" ypos = "74"`<br>`</INIT>`<br><br>`<PAINT>`<br>`<TEXT xpos = "23" ypos = "34">R - Parameter 1</TEXT>`<br>`<TEXT xpos = "23" ypos = "54">R - Parameter 2</TEXT>`<br>`<TEXT xpos = "23" ypos = "74">R - Parameter 3</TEXT>`<br><br>`</PAINT>`<br><br>`</FORM>` |
| **INIT** | **Dialog box message**<br>The tag is executed immediately after the dialog box is generated. All the input elements and hotlinks for the dialog form should be created here. |

| Tag identifier | Description |
|---|---|
| **KEY_EVENT** | Dialog message<br>The tag KEY_EVENT can be integrated in the form to evaluate keyboard events. The system sends the MF2 keyboard code to the active form if the tag is available in a form. If the variable **$actionresult** is not set to zero, the system then subsequently processes the keyboard event.<br>The keyboard code is provided in the variable **$keycode** as an integer value.<br><br>**Example:**<br>The character entered into the variable **exclude_key** should be filtered-out of the input stream.<br><br>`<LET name="stream" type="string"/>`<br>`<LET name="exclude_key" type="string"/>`<br><br>`<FORM name = "keytest_form">`<br>`  <INIT>`<br>`  <CONTROL name = "p1" xpos = "120" ypos = "84" width ="200" refvar="stream" hotlink="true" />`<br>`  <CONTROL name = "p2" xpos = "160" ypos = "104" width ="8" refvar="exclude_key" hotlink="true" />`<br>`  </INIT>`<br><br>`  <PAINT>`<br>`  <text xpos = "8" ypos = "84">data stream</text>`<br>`  <text xpos = "8" ypos = "104">exclude key</text>`<br>`  </PAINT>`<br><br>`  <KEY_EVENT>`<br>`  <LET name="excl_keycode" type="string"/>`<br>`  <OP>excl_keycode = exclude_key</OP>`<br>`  <type_cast name="excl_keycode" type="int" />`<br>`  <PRINT text="%d %d">$keycode, excl_keycode</PRINT>`<br><br>`  <IF>`<br>`  <CONDITION>$keycode == excl_keycode</CONDITION>`<br>`  <THEN>`<br>`  <op> $actionresult = 0</op>`<br>`  </THEN>`<br>`  </IF>`<br><br>`  </KEY_EVENT>`<br><br>`</FORM>` |

| Tag identifier | Description |
|---|---|
| **MOUSE_EVENT** | The tag can be linked into the script for processing mouse events. It is executed when the following activities have been performed with the mouse:<br>• A button has been pressed<br>• A button has been released<br>• The mouse has been moved<br>The parser provides the information in a structure and creates the structure variable **$mouse_event** with the following elements:<br><br>**Structure elements:**<br>• **type**<br>Coding of the activity<br>– 2 - A button has been pressed<br>– 3 - A button has been released<br>– 5 - The mouse has been moved<br>• **x**<br>X-position of the cursor in pixels; relative to the current screen resolution<br>• **y**<br>Y-position of the cursor in pixels; relative to the current screen resolution<br>• **id**<br>Identifier<br>– -1, if the position cannot be assigned to any control<br>– != -1, if the mouse cursor is inside a control, the content of attribute **idemdata** is returned<br>• **button**<br>Contains the state of the buttons at the time of the event<br>– 0 - No button<br>– 1 - Left button<br>– 2 - Right button<br>– 4 - Center button<br>The buttons can be associated with a bit-by-bit OR operation.<br><br>**Example:**<br>`<MOUSE_EVENT>`<br>`<print text="button %d type %d x %d y %d ">$mouse_event.button, $mouse_event.type, $mouse_event.x, $mouse_event.y</print>`<br>`</MOUSE_EVENT>` |

| Tag identifier | Description |
|---|---|
| MESSAGE | **Dialog message**<br>If the Send_message operation is executed in the script, then the parser processes the tag message. Values P1 and P2 are provided in the variables $message_par1 and $message_par2 (see the "SEND_MESSAGE" tag).<br><br>**Syntax:**<br>`<MESSAGE>`<br>`</MESSAGE>`<br><br>**Example:**<br>`<LET name="user_selection" />`<br><br>`<SOFTKEY POSITION="3">`<br>`<CAPTION>Set%nParameter</CAPTION>`<br>`<SEND_MESSAGE>1, 10</SEND_MESSAGE>`<br>`</SOFTKEY>`<br>…<br>…<br><br>`<FORM>`<br>…<br>…<br>`<MESSAGE>`<br>`<SWITCH>`<br>`<CONDITION>$message_par1</CONDITION>`<br>`<CASE value="1">`<br>`<OP> user_selection = $message_par2 </OP>`<br>…<br>…<br>`</CASE>`<br>`</SWITCH>`<br>`</MESSAGE>`<br>…<br>…<br>`</FORM>` |

| Tag identifier | Description |
|---|---|
| **SEND_MESSAGE** | The tag sends a message with two parameters to the active form, which is processed in the tag message (see also MESSAGE).<br><br>**Syntax:**<br>`<SEND_MESSAGE>p1, p2</SEND_MESSAGE>`<br><br>**Example:**<br>`<LET name="user_selection" />`<br><br>`<SOFTKEY POSITION="3">`<br>`  <CAPTION>Set%nParameter</CAPTION>`<br>`  <SEND_MESSAGE>1, 10</SEND_MESSAGE>`<br>`</SOFTKEY>`<br>…<br>…<br>`<FORM>`<br>…<br>…<br>`<MESSAGE>`<br>`<SWITCH>`<br>`<CONDITION>$message_par1</CONDITION>`<br>`<CASE value="1">`<br>`<OP> user_selection = $message_par2 </OP>`<br>…<br>`</CASE>`<br>`</SWITCH>`<br>`</MESSAGE>`<br>…<br>…<br>`</FORM>` |

| Tag identifier | Description |
|---|---|
| RESIZE | **Dialog box message**<br>The tag can be linked into the script for processing a RESIZE event. This event is created by a dynamic resolution switchover.<br> |

| Tag identifier | Description |
|---|---|
| **RESIZE** Continued | **Example:**<br>`<let name="screen_size" type="StructSize" />`<br>`<form name="menu_userscale_form" autoscale_content="off">`<br>` <init>`<br>` <caption>Use auto scaling</caption>`<br><br>` <function name="hmi.get_hmi_resolution">screen_size</function>`<br>` <print text="screen size: w %d h: %d">screen_size.width, screen_size.height</print>`<br><br><br>` <data_access type="true" />`<br>` <control name="s_c" xpos="8" ypos="140" fieldtype="readonly" width="500" refvar="string_var" hotlink="true"/>`<br>` <if>`<br>` <condition>screen_size.width == 800</condition>`<br>` <then>`<br>` <control name="s_1" xpos="508" ypos="140" fieldtype="readonly" width="100" refvar="string_var1" hotlink="true"/>`<br>` </then>`<br>` </if>`<br>` <control name="s_w" xpos="200" ypos="350" fieldtype="readonly" refvar="screen_size.width" hotlink="true"/>`<br>` <control name="s_h" xpos="200" ypos="370" fieldtype="readonly" refvar="screen_size.height" hotlink="true"/>`<br>` </init>`<br>` <paint>`<br>` <text xpos ="68" ypos="310">Text</text>`<br>` <text xpos ="8" ypos="350">screen width</text>`<br>` <text xpos ="8" ypos="370">screen height</text>`<br>` </paint>`<br><br>` <resize>`<br>` <function name="hmi.get_hmi_resolution">screen_size</function>`<br>` <print text="resize_event screen size: w %d h: %d">screen_size.width, screen_size.height</print>`<br>` <switch>`<br>` <condition>screen_size.width</condition>`<br>` <case value="640">`<br>` <function name="control.delete">_T"s_1"</function>`<br>` </case>`<br>` <case value="800">`<br>` <control name="s_1" xpos="508" ypos="140" fieldtype="readonly" width="100" refvar="string_var1" hotlink="true"/>`<br>` </case>`<br>` </switch>`<br>` </resize>`<br>`</form>` |

| Tag identifier | Description |
|---|---|
| **FOCUS_IN** | **Dialog box message**<br>The tag is called if the system places the focus on a control. To identify the control, the system copies the name of the control into variable **$focus_name** and the value of the attribute **item_data** into variable **$focus_item_data**.<br>This message can be used, e.g. to output images depending on the focus position.<br>**Example:**<br>`<focus_in>`<br>`<PRINT text="focus on filed:%s, %d">$focus_name,`<br>`$focus_item_data </PRINT>`<br>`</focus_in>` |
| **PAINT** | **Dialog box message**<br>The tag is executed when the dialog box is displayed. All the texts and images which are to be displayed in the dialog box should be specified here.<br>Further, the tag is executed if the system identifies that parts of the dialog box are to be redisplayed. For example, this can be initiated by closing high-level windows. |
| **TIMER** | **Dialog box message**<br>The tag is executed cyclically.<br>Each form is assigned a timer that initiates that the timer - tag is executed approx. every 100 ms. |
| **CAPTION** | The tag contains the title of the dialog box.<br>This tag should be used within the INIT tag.<br>The title bar can be divided into multiple columns.<br>To divide the title bar, the "caption" tag must be programmed with the "define_section" attribute before the text is output.<br>The "define_section" attribute specifies the number of columns<br><br>The "property" attribute is used to define the length, starting position, and text alignment for each column. The position and length specification is a percentage value referring to the width of the form.<br>The value of the "value" attribute indicates the column number (beginning with zero)<br><br>`<caption define_sections="3">`<br>`<property value="0" length="20" position="2" alignment="left" />`<br>`<property value="1" length="20" position="79"`<br>`alignment="right" />`<br>`</caption>`<br><br>The text can then be assigned to the column by additionally specifying the "index" attribute with the column index.<br>`<caption index="0">column1</caption>`<br>`<caption index="1">column2</caption>`<br><br>**Syntax:**<br>`<CAPTION>Titel</CAPTION>`<br><br>**Example:**<br>`<CAPTION>my first dialogue</CAPTION>` |
| **CLOSE** | **Dialog box message**<br>This tag is executed before the dialog box is closed. |

| Tag identifier | Description |
|---|---|
| CLOSE_FORM | The tag closes the active dialog.<br>This instruction is only necessary if the dialog is opened by the MMC command and the user is offered a softkey function to close the dialog. Generally, dialogs are automatically managed and do not have to be explicitly closed.<br><br>**Syntax:**<br>`<CLOSE_FORM/>`<br><br>**Example:**<br>`<softkey_ok>`<br>`<caption>OK</caption>`<br>`<CLOSE_FORM />`<br>`<navigation>main_menu</navigation>`<br>`</softkey_ok>` |

| Tag identifier | Description |
|---|---|
| CONTROL | The tag is used to generate control elements.<br>**Syntax:**<br>`<CONTROL name = "<control name>" xpos = "<X position>" ypos = "<Y position>" refvar = "<NC variable>" hotlink = "true" format = "<format>" />`<br><br>**Note:**<br>Variables used to set attribute values must be declared global variables.<br><br>**Attributes:**<br>• **name**<br>Identifier of the field.<br>The identifier simultaneously represents a local variable, and must not be used a multiple number of times in the form.<br>• **xpos**<br>X position of the top left corner<br>• **ypos**<br>Y position of the top left corner<br>• **fieldtype**<br>Field type<br>If no type is specified, the field is set as an edit field.<br>– **edit**<br>Data can be changed<br>– **readonly**<br>Data cannot be changed<br>**combobox**<br>The field displays the corresponding identifiers instead of numerical values.<br>If field type **combobox** is selected, the expressions to be displayed must also be assigned to the field.<br>The <ITEM> tag should be used for this purpose.<br>The combo box saves the index of the currently selected text in the variable belonging to the control (see the attribute **refvar**).<br>After the control has been created, additional elements can be inserted using the functions **addItem** or **insertItem**.<br>– **progressbar**<br>A progress bar with a value range of 0 to 100 appears.<br>The valley value and peak value properties can be used to adapt the value range to the data to be displayed. |

| Tag identifier | Description |
|---|---|
| **CONTROL** Continued | • **fieldtype**<br>– **graphicbox**<br>The field type generates a 2D broken line graphic control. Using the tag <ITEM> a graphical element can be inserted into the control. Parameters **width** and **height** specify the width and height of the box.<br>After the control has been created, additional elements can be inserted using the functions **addItem** or **insertItem**. Parameter **itemdata** is not evaluated for this control.<br>If a reference variable is assigned to the control, its values will be recorded in a 100 ms time base. Up to 10,000 values can be recorded. With the property **max**, values are recorded endlessly. The oldest value is then deleted when the maximum recording time is reached. The recording time must be specified in milliseconds.<br>Although the values are stored at discrete times, the control displays them as connected lines. The attribute **channel** permits recording of up to three further reference variables. The index starts with one. Index zero is used to set the properties of the variables assigned in the control tag.<br>**Example:**<br>`<control name= "c_gbox1" xpos = "250" ypos="24" width="240" height="356" fieldtype="graphicbox" refvar="val" hotlink="true" COLOR_BK="#ffffff">`<br>`  <property max="60000" /> <!-- steps of ms -->`<br>`  <property min="0" />`<br>`  <property channel="0" color="#00ffff" penwidth="3"/>`<br>`  <property ORDINATE="Y sinus" />`<br>`  <property ABSCISSA="time *100 ms" />`<br>`  <property channel="1" refvar="val2" color="#000000" />`<br>`</control>` |

| Tag identifier | Description |
|---|---|
| **CONTROL** Continued | **Example graphicbox:**<br>`<CONTROL name= "graphic" xpos = "8" ypos="23" width="300" height="352" fieldtype="graphicbox" />`<br>• Adding elements:<br>Elements are added using the function **additem** or **loaditem**.<br>The following 2d elements can be used:<br>– Line - l(inc)<br>– Circle sector - c(ircle)<br>– Point - p(oint)<br>**Structure of an element:**<br><Element type>; coordinates<br>• Line:<br>l; xs; ys; xe, ye<br>l - line marking<br>Xs - X start position<br>Ys - Y start position<br>Xe - X end position<br>Ye - Y end position<br>• Circle:<br>C, xs, ys, xe, ye, cc_x, cc_y, r<br>C - circular sector marking<br>Xs - X start position<br>Ys - Y start position<br>Xe - X end position<br>Ye - Y end position<br>Cc_x - X coordinate, circle center point<br>CC_y - Y coordinate circle center point<br>• Radius:<br>R<br>• Point:<br>P, x, y<br>P - point marking<br>X - X position<br>Y - Y position<br>• Deleting the graphic:<br>The content is deleted using the function **empty**. |

| Tag identifier | Description |
|---|---|
| **CONTROL** Continued | The following field types are described in section "Configuring your own buttons (Page 161)".<br>• **fieldtype**<br>– **pushbutton**<br>The field type is used as a pushbutton or a latching button.<br>– **radiobutton**<br>The field type allows you to select one of several options.<br>– **checkbox**<br>The field type allows you to select several options.<br>– **groupbox**<br>The field type visually encloses a group of controls with a frame and a title.<br>– **switch**<br>The field type is a graphical element that signals one of two states using an icon.<br>– **scrollarea**<br>The field type is used to display controls within a specified area. |
| **CONTROL** Continued | • **fieldtype**<br>– **listbox**<br>The field type generates an empty list box control.<br>Using the tag <ITEM> a list box element can be inserted in the list box.<br>The ITEM attribute **value** allows this element to be assigned a unique value.<br>This can serve to identify the element, for example.<br>Parameters **width** and **height** specify the width and height of the list box.<br>After the control has been created, additional list box elements can be inserted using the functions **addItem**, **insertItem**, or **loadItem**.<br>After the control has been created, additional list box elements can be inserted using the function **addItem** or **insertItem**. Parameter **itemdata** is not evaluated for this control.<br>– **itemlist**<br>The field type generates a static control, which displays the corresponding identifier instead of numerical values.<br>The <ITEM> tag can be used to assign an identifier to the field.<br>• **item_data**<br>A user-specific integer value can be assigned to the attribute. This value is given as part of the FOCUS_IN message for identifying the focus field.<br>• **refvar**<br>Identifier of the reference variable that can be linked to the field (optional).<br>• **hotlink** = "TRUE" " If the value of the reference variable changes, then the field is automatically updated (optional).<br>• **format**<br>The attribute defines the display format of the specified variable.<br>Formatting data, see print-Tag (optional). |

| Tag identifier | Description |
|---|---|
| **CONTROL** Continued | **Attributes:**<br>• **color_bk**<br>The attribute sets the background color of the control.<br>• **color_fg**<br>The attribute sets the foreground color of the control.<br>Color coding (see Chapter "Color coding (Page 48)")<br>• **display_format**<br>The attribute defines the processing format of the specified variable. This attribute must be used when accessing a PLC float variable, as the access is realized by reading a double word.<br>The following data formats are permitted:<br>– FLOAT<br>– INT<br>– DOUBLE<br>– STRING<br>Assigning expressions (e.g. text or graphic element to be displayed) to a list box, graphics box or combo box:<br>**Syntax:**<br><ITEM>Expression</ITEM><br><ITEM value ="<Value>">Expression</ITEM> |

| Tag identifier | Description |
|---|---|
| **CONTROL** Continued | **Example:**<br>`<CONTROL name = "button1" xpos = "10" ypos = "10" fieldtype = "combobox ">`<br>`<ITEM>text1</ITEM>`<br>`<ITEM>text2</ITEM>`<br>`<ITEM>text3</ITEM>`<br>`<ITEM>text4</ITEM>`<br>`</CONTROL>`<br>If any integer value is to be assigned to an expression, the attribute **value = "value"** should be added to the tag.<br>Rather than consecutive numbers, the control variable now contains the item's assigned value.<br><br>**Example:**<br>`<CONTROL name = "button1" xpos = "10" ypos = "10" fieldtype = "combobox ">`<br>`<ITEM value = "10">text1</ITEM>`<br>`<ITEM value = "20">text2</ITEM>`<br>`<ITEM value = "12">text3</ITEM>`<br>`<ITEM value = "1">text4</ITEM>`<br></CONTROL><br>**Example of a progress bar:**<br>`<CONTROL name = "progress1" xpos = "10" ypos = "10" width = "100" fieldtype = "progressbar" hotlink = "true" refvar = "nck/Channel/GeometricAxis/actProgPos[1]">`<br>`<PROPERTY min = "0" />`<br>`<PROPERTY max = "1000" />`<br>`</CONTROL>`<br>**Example, list box:**<br>`<let name="item_string" type="string"></let>`<br>`<let name="item_data" ></let>`<br><br>`<CONTROL name="listbox1" xpos = "360" ypos="150" width="200" height="200" fieldtype="listbox" />`<br>• Adding elements:<br>Elements are added using the function **additem** or **loaditem**.<br>• Deleting the content:<br>The content is deleted using the function **empty**.<br><br>`<op> item_string = _T"text1\\n" </op>`<br>`<function name="control.additem">_T"listbox1", item_string, item_data </function>`<br>`<op> item_string = _T"text2\\n" </op>`<br>`<function name="control.additem">_T"listbox1", item_string, item_data </function>` |
| **CONTROL** Continued | **Example itemlist:**<br>`<CONTROL name = "itemlist1" xpos = "10" ypos = "10" fieldtype = "itemlist">`<br>`<ITEM value = "10">text1</ITEM>`<br>`<ITEM value = "20">text2</ITEM>`<br>`<ITEM value = "12">text3</ITEM>`<br>`<ITEM value = "1">text4</ITEM>`<br>`</CONTROL>` |

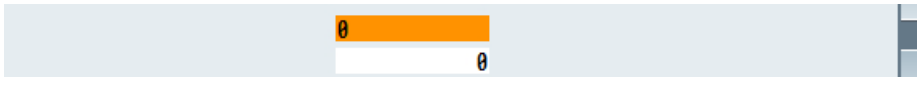
| Tag identifier | Description |
|---|---|
| **CONTROL** Continued | The image box control manages a picture in bitmap or GIF format. If the picture is larger than the displayable area, the control will show a scroll bar.<br>To control the visible area, the system provides the function CONTROL.IMAGEBOXSET (see Chapter "Predefined functions (Page 107)").<br><br>**Syntax:**<br>`<function name="control.imageboxget"> ... </function>` |
| **CONTROL** Continued | **Attributes:**<br>• **disable**<br>The attribute locks/permits the input in an edit control.<br>• **tooltip**<br>An information text is displayed if the cursor is placed on the control.<br>• **factor**<br>Conversion factor<br>• **font**<br>Definition of a font |
| **CONTROL** Continued | **Changing the control after creation**<br>A control tag changes the properties of an existing control after it has been created. The tag must be specified with the name of the control to be changed and the new properties. It can be executed only within a form tag. The following properties can be changed, for example:<br>• name<br>• xpos<br>• ypos<br>• width<br>• height<br>• color_bk<br>• color_fg<br>• access level<br>• fieldtype<br>• itemdata<br>• min<br>• max<br>• default<br>• disable<br>• tooltip<br>• font<br>• factor<br>The reference variable cannot be modified. If a property is to be changed by triggering by a softkey event, the send message tag must transfer this request into the form context. The message tag is used to acquire the message. |

| Tag identifier | Description |
|---|---|
| **CONTROL** Continued | **Control change in an operating instruction**<br>Another possibility of changing properties of a control during runtime is to make the change in an operating instruction. Therefore the name of the control and the property to which a new value is to be assigned must be defined. The property is separated from the control name by a point.<br>**Syntax:**<br><Name>.<Property><br><br>**Example:**<br>`...`<br>`...`<br>`<let name="value" />`<br>`<let name="w" />`<br>`<let name="h" />`<br>`...`<br>`...`<br>`<control name="c_move" xpos="$xpos" ypos="124" />`<br>`...`<br>`...`<br>`<op>`<br>`c_move.xpos = 300;`<br>`value = c_move.xpos;`<br>`h = c_move.height;`<br>`w = c_move.width;`<br>`</op>` |
| **HELP_CONTEXT** | This tag defines the help topic to be called. It should be programmed in the INIT block.<br>The name specified in the attribute is supplemented by the prefix XmlUserDlg_ and is transferred to the help system. The associated structure of the help file should be taken from the topic - generating an online help.<br>**Sequence when activating the help system:**<br>1. Press the "Info" key.<br>2. The dialog supplies the expression "my_dlg_help".<br>3. Parser converts the expression into "XmlUserDlg_my_dlg_help" .<br>4. Activating the help system.<br>5. Submitting the search term "XmlUserDlg_my_dlg_help".<br><br>**Syntax:**<br>`<HELP_CONTEXT name="<context name>" />`<br><br>**Example:**<br>`...`<br>`<INIT>`<br>`...`<br>`<CAPTION>my dialogue</CAPTION>`<br>`<HELP_CONTEXT name="my_dlg_help" />`<br>`...`<br>`</INIT>` |

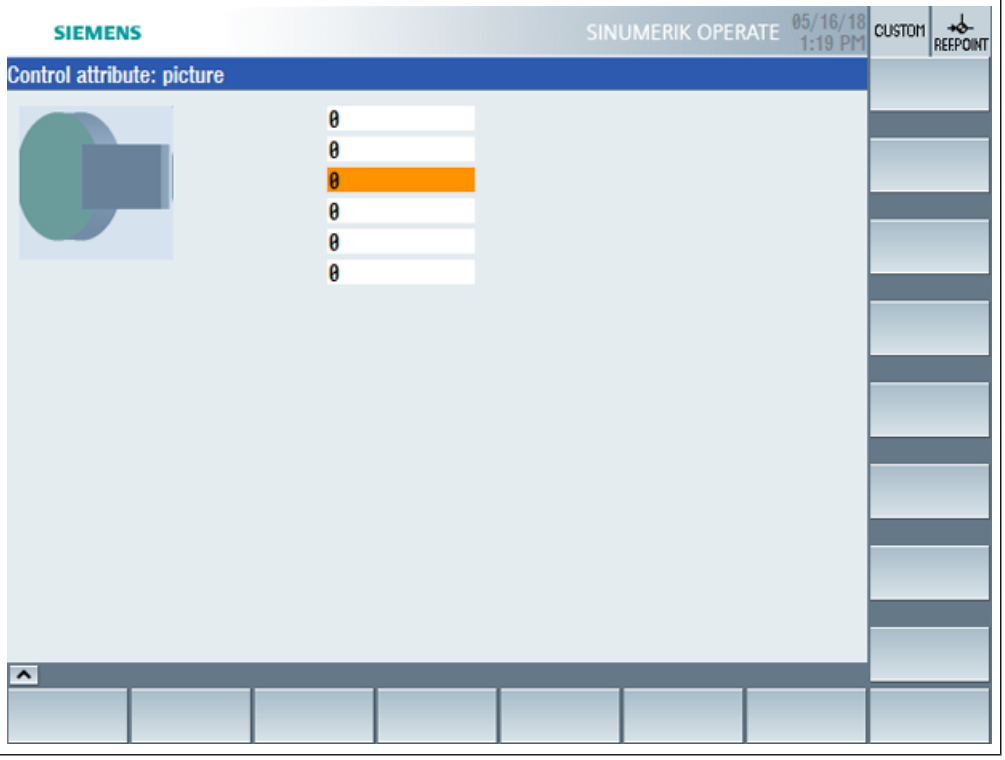
| Tag identifier | Description |
|---|---|
| DATA_ACCESS | The tag controls the behavior of the dialog forms when user inputs are being saved.<br>The behavior should be defined within the INIT tag.<br>If the tag is not used, inputs are buffered in each case.<br>Exception: Controls for which the **hotlink** attribute is set to **true** are always written to and read directly.<br><br>**Attribute:**<br>• **type** = "TRUE" – the input values are not buffered. The dialog form copies the input values to the reference variables directly.<br>• **type =** "FALSE" – the values are only copied to the reference variable with the **UPDATA_DATA** type = "FALSE" tag<br><br>**Example:**<br>`<DATA_ACCESS type = "true" />` |
| EDIT_CHANGED | **Dialog box message**<br>This tag is called if the contents of an edit control have changed.<br>To identify the control, the system copies the name of the control into variable **$focus_name** and the value of the attribute **item_data** into variable **$focus_item_data**.<br><br>**Example:**<br>`<EDIT_CHANGED>`<br>`<print text="index changed filed:%s, %d"> $focus_name, $focus_item_data </print>`<br>…<br>…<br>`</EDIT_CHANGED>` |
| GESTURE_EVENT | The tag is used to perform finger gestures for multi-touch operation.<br>The tag is described in section "Multitouch operation (Page 153)". |
| INDEX_CHANGED | **Dialog box message**<br>The tag is called, if the operator changes the selection of a combo box.<br>To identify the control, the system copies the name of the control into the variable **$focus_name** and the value of the attribute **item_data** into the variable **$focus_item_data**.<br>**Note:**<br>A reference variable assigned to the control, has not been aligned to the control variable at this point in time and contains the index of the previous selection of the combo box.<br><br>**Example:**<br>`<INDEX_CHANGED>`<br>`<print text="index changed filed:%s, %d"> $focus_name, $focus_item_data </print>`<br>…<br>…<br>`</INDEX_CHANGED>` |

| Tag identifier | Description |
|---|---|
| **MENU** | The tag defines a menu containing the softkey description and the dialog to be opened.<br><br>**Attribute:**<br>• **name**<br>Menu name<br><br>**Syntax:**<br>`<MENU name = "<menu name>">`<br>…<br>`<open_form …>`<br>…<br>`<SOFTKEY …>`<br>`</SOFTKEY>`<br>`</MENU>` |
| **NAVIGATION** | This tag defines the menu to be called. It can be used within a softkey block, a menu block, and in a form. If a variable name is assigned to the tag as its value, the parser will activate the menu stored in the variable.<br>In a menu block, the navigation is at the position in the instruction. Subsequent instructions are no longer executed.<br><br>**Note:**<br>If the navigation target is to be determined by the content of a variable, this variable must be declared a global variable.<br><br>**Syntax:**<br>`<NAVIGATION>menu name</NAVIGATION>`<br><br>**Example:**<br>`<menu name = "main">`<br>`<softkey POSITION="1">`<br>`<caption>sec. form</caption>`<br>`<navigation>sec_menu</navigation>`<br>`</softkey>`<br>`</menu>`<br><br>`<menu name = "sec_menu">`<br>`<open_form name = "sec_form" />`<br>`<softkey_back>`<br>`<navigation>main</navigation>`<br>`</softkey_back>`<br>**</menu>** |

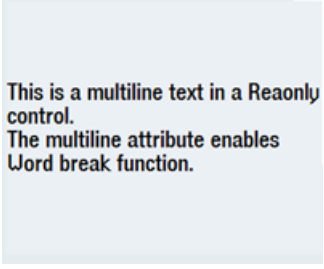
| Tag identifier | Description |
|---|---|
| **OPEN_FORM** | The tag opens the dialog form given under the name.<br><br>**Attribute:**<br>• **name**<br>Name of the dialog form<br><br>**Syntax:**<br>`<OPEN_FORM name = "<form name>" />`<br><br>**Example:**<br>`<menu name = "main">`<br>`<open_form name = "main_form" />`<br>`<softkey POSITION="1">`<br>`<caption>main form</caption>`<br>`<navigation>main</navigation>`<br>`</softkey>`<br>`</menu>`<br><br>`<form name="main_form">`<br>`<init>`<br>`</init>`<br><br>`<paint>`<br>`</paint>`<br><br>`</form>` |

| Tag identifier | Description |
|---|---|
| **PROPERTY** | This tag can be used to define additional properties for an operator control. The tag is embedded in the control tag.<br><br>**Attributes:**<br>• **max** = "<maximum value>"<br>• **min** = "<minimum value>"<br>• **default** = "<pre-assignment>"<br>• **factor** = "conversion factor"<br>• **color_bk** = "<background color coding>"<br>• **color_fg** = "<font color coding>"<br>• **password** = "<true>" - entered character is displayed with "*"<br>• **multiline** = "<true>" - permits multi-line inputs in an edit control<br>• **disable** = "<true/false>" - locks/permits the input in an edit control<br>• **transparent** = "Transparent color of a bitmap"<br>Color coding (see Chapter "Color coding (Page 48)")<br>• **tooltip** = information text is displayed if the cursor is set to the control.<br>The syntax is: <property tooltip="note text" /><br>• **abscissa** = "Name of the first coordinate axis" (only permissible for a graphic box)<br>• **ordinate** = "Name of the second coordinate axis" (only permissible for a graphic box)<br>**Example:**<br>`<CONTROL name = "progress1" xpos = "10" ypos = "10" width = "100" fieldtype = "progressbar" hotlink = "true" refvar = "nck/Channel/GeometricAxis/actProgPos[1]">`<br>`<PROPERTY min = "0" />`<br>`<PROPERTY max = "1000" />`<br>`</CONTROL>`<br>`<CONTROL name = "edit1" xpos = "10" ypos = "10">`<br>`<PROPERTY min = "20" />`<br>`<PROPERTY max = "40" />`<br>`<PROPERTY default = "25" />`<br>`</CONTROL>` |
| **PROPERTY** continued | **Attributes:**<br>**alignment** - this attribute defines the text alignment for entry fields<br><br>**Syntax:**<br>`alignment="<right/left>"`<br><br>**Example:**<br>`<control name="edit2" xpos="208" ypos="52" >`<br>`<property alignment="right"/>`<br>`</control>`<br><br><br> |

| Tag identifier | Description |
|---|---|
| **PROPERTY** continued | **Attributes:**<br>**picture** - this attribute defines a graphic to be displayed<br><br>The specified graphic is displayed or the name of the graphic is stored in a variable if the input focus is set on this field. This attribute is to be used in conjunction with the **refvar** attribute to be able to define the data sink.<br>To display a graphic or an animation, the name of an **imagebox** type control must be specified. This control assumes management of the graphic.<br>If the name of a local variable is specified, this variable must be of a **String** data type.<br>The graphic continues to be displayed until a new name is assigned to the reference variables.<br><br>**Syntax:**<br>`<property picture="<Name der Grafik>" refvar="<Datensenke>"/>` |

| Tag identifier | Description |
|---|---|
| **PROPERTY** continued | **Example:**<br>`<let name="cursor_icon" type="string" />`<br>` <form name="main_form1">`<br>`  <init>`<br>`  <caption>Control attribute: picture</caption>`<br>`  <control name="cursor_image" xpos="8" ypos="32" width="100" height="100" fieldtype="imagebox" />`<br>`  <control name="edit1" xpos="208" ypos="32" >`<br>`  <property picture="red_led_off.bmp" refvar="cursor_image"/>`<br>`  </control>`<br>`  <control name="edit2" xpos="208" ypos="52" >`<br>`  <property picture="red_led_on.bmp" refvar="cursor_image"/>`<br>`  </control>`<br>`  <control name="edit3" xpos="208" ypos="72" >`<br>`  <property picture="test.gif" refvar="cursor_image"/>`<br>`  </control>`<br>`  <control name="edit4" xpos="208" ypos="92" >`<br>`  <property picture="red_led_off.bmp" refvar="cursor_image"/>`<br>`  </control>`<br>`  <control name="edit5" xpos="208" ypos="112" >`<br>`  <property picture="red_led_off.bmp" refvar="cursor_icon"/>`<br>`  </control>`<br>`  <control name="edit6" xpos="208" ypos="132" >`<br>`  <property picture="red_led_on.bmp" refvar="cursor_icon"/>`<br>`  </control>`<br>`  </init>`<br>`  <timer><print text="%s">cursor_icon</print></timer>`<br>` </form>`<br> |

| Tag identifier | Description |
|---|---|
| **PROPERTY** continued | **Attributes:**<br>**cursortext** - this attribute defines the cursor text to be displayed<br><br>The text is issued in the title bar if the input focus is set on this field.<br>Two further attributes can be specified in addition to this attribute which define the alignment of the text and the cursor text area. The text is left-justified as standard. The cursor text area takes up 50% of the title bar width by default.<br><br>`alignment="<right/left>"` - Text alignment right-justified/left-justified<br>`length="<Width>"` - percentage of the title bar which should be required for the cursor text<br><br>**Syntax:**<br>`<property cursortext="<text>" />`<br>or<br>`<property cursortext="<text>" alignment="<right/left>" />`<br>or<br>`<property cursortext="text2" alignment="r="<text>" alignment="<right/left>" length="<Ratio>"/>`<br><br>**Example:**<br>`<form name="main_form2">`<br>` <init>`<br>`  <caption>Control attribute: cursortext</caption>`<br>`  <control name="edit1" xpos="208" ypos="32" >`<br>`  <property cursortext="cursor text field edit1" />`<br>`  </control>`<br>`  <control name="edit2" xpos="208" ypos="52" >`<br>`  <property cursortext="cursor text field edit2" alignment="right"/>`<br>`  </control>`<br>`  <control name="edit3" xpos="208" ypos="72" >`<br>`  <property cursortext="cursor text field edit3" alignment="right" length="40"/>`<br>`  </control>`<br>`  <control name="edit4" xpos="208" ypos="92" >`<br>`  <property cursortext="cursor text field edit4" />`<br>`  </control>`<br>` </init>`<br>`</form>`<br><br> |

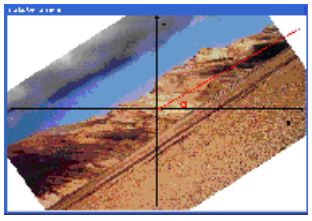
| Tag identifier | Description |
|---|---|
| **PROPERTY** continued | **Attributes:**<br>**multiline** - this allows the multi-line output of a text in a read-only control<br><br>**Syntax:**<br>`<property multiline="true" />`<br><br>**Example:**<br>`...`<br>`...`<br>`<op>`<br>`textlist[2] = _T"This is a multiline text in a Reaonly control.\`<br>`\nThe multiline attribute enables Word break function.";`<br>`</op>`<br>`<control name="lstring" xpos="8" ypos="120" width="200" height="160"`<br>`fieldtype="readonly" refvar="textlist[2]" >`<br>`<property multiline="true" />`<br>`</control>`<br><br> |
| **SOFTKEY** | The tag defines the properties and responses of a softkey.<br>**Attributes:**<br>• **position**<br>Number of the softkey. 1-8 horizontal softkeys, 9-16 vertical softkeys<br>The following attributes become effective from:<br>• **type**<br>Defines the property of the softkey.<br>user_controled - The script defines how the softkey is displayed<br>toggle_softkey - The softkey is displayed alternating between pressed and not pressed<br>• **refvar**<br>Should only be used in conjunction with toggle_softkey .<br>Reference variable, into which the actual softkey property is copied.<br>A variable, type "String" should be specified, which includes the properties pressed, not pressed or locked (see tag state).<br>• **picture**<br>Using the attribute, a bitmap can be output left justified on the softkey. The complete path name should be specified.<br>The number of text characters that can be displayed is reduced to the width of the bitmap. |

| Tag identifier | Description |
|---|---|
| **SOFTKEY** continued | The following additional actions can be defined within the softkey block:<br>• **picturealignment**<br>The image orientation is specified by this attribute.<br>The image is aligned with the left side of the softkey by default.<br>The following values can be specified for alignment:<br>– **top**<br>Top edge<br>– **bottom**<br>Bottom edge<br>– **left**<br>Left edge<br>– **right**<br>Right edge<br>– **center**<br>Centered<br>• **caption**<br>Softkey text<br>• **state**<br>Should only be used in conjunction with user_controlled .<br>The tag assigns the required softkey display to the system.<br>**Syntax:**<br><state type="<state>" /><br>The following strings can be specified:<br>– **notpressed**<br>The softkey is displayed as being not pressed.<br>– **pressed**<br>The softkey is displayed as being pressed.<br>– **disabled**<br>The softkey is locked and is displayed in gray.<br>• **navigation**<br>• **update_controls**<br>• **function** |

| Tag identifier | Description |
|---|---|
| **SOFTKEY** continued | **Syntax:**<br>Standard softkey:<br>`<state type="<softkey state>" />`<br>`<softkey position = "<1>">`<br>…<br>…<br>`</softkey>`<br>or<br>Script-controlled softkey:<br>`<softkey position = "<1>" type="<user_defined>" >`<br>`<state type="<softkey state>" />`<br>…<br>…<br>`</softkey>`<br><br>or<br>Toggle softkey:<br>`<softkey position = "<1>" type="<toggle_softkey>" refvar="<variable name>" >`<br>…<br>…<br>`</softkey>`<br><br>**Example:**<br><br>`<let name="define_sk_type" type="string">PRESSED</let>`<br>`<let name="sk_type">1</let>`<br><br>`<softkey POSITION="1" type="user_controled" >`<br>`<caption>Toggle%nSK</caption>`<br>`<if>`<br>`<condition>sk_type == 0 </condition>`<br>`<then>`<br>`<op> sk_type = 1 </op>`<br>`<op> define_sk_type = _T"PRESSED" </op>`<br>`</then>`<br>`<else>`<br>`<op> define_sk_type = _T"NOTPRESSED" </op>`<br>`<op> sk_type = 0 </op>`<br>`</else>`<br>`</if>`<br>`<state type="$$$define_sk_type" />`<br>…<br>…<br><br>`</softkey>` |

| Tag identifier | Description |
|---|---|
| **SOFTKEY** continued | **Example:**<br>or<br>`<let name="curr_softkey_state" type="string">PRESSED</let>`<br>`</softkey>`<br>`<softkey POSITION="3" type="toggle_softkey" refvar="curr_softkey_state">`<br>`<caption>Toggle%nSK</caption>`<br>…<br>**</softkey>**<br><br><br><br> |
| **SOFTKEY_OK** | The tag defines the response of the softkey "OK".<br><br><br>The following additional actions can be defined within the softkey block:<br>• **navigation**<br>• **update_controls**<br>• **function**<br>**Syntax:**<br>`<SOFTKEY_OK>`<br>…<br>…<br>`</SOFTKEY_OK>` |
| **SOFTKEY_CANCEL** | The tag defines the response of the softkey "Cancel".<br><br><br>The following additional actions can be defined within the softkey block:<br>• **navigation**<br>• **update_controls**<br>• **function**<br>**Syntax:**<br>`<SOFTKEY_CANCEL>`<br>…<br>…<br>`</SOFTKEY_CANCEL>` |

| Tag identifier | Description |
|---|---|
| **SOFTKEY_BACK** | The tag defines the response of the softkey "Back".<br><br><br>The following additional actions can be defined within the softkey block:<br>• **navigation**<br>• **update_controls**<br>• **function**<br>**Syntax:**<br>`<SOFTKEY_BACK>`<br>…<br>…<br>`</SOFTKEY_BACK>` |
| **SOFTKEY_ACCEPT** | The tag defines the response of the softkey "Accept".<br><br><br>The following additional actions can be defined within the softkey block:<br>• **navigation**<br>• **update_controls**<br>• **function**<br>**Syntax:**<br>`<SOFTKEY_ACCEPT>`<br>…<br>…<br>`</SOFTKEY_ACCEPT>` |
| **TEXT** | The tag is used to display a text in the specified position.<br>If an alarm number is used, the dialog box displays the text which is saved for the number.<br>**Syntax:**<br>`<TEXT xpos = "<X position>" ypos = "<Y position>"> Text </TEXT>`<br>**Attributes:**<br>• **xpos**<br>X position of the top left corner<br>• **ypos**<br>Y position of the top left corner<br>• **color**<br>Text color (see Chapter "Color coding (Page 48)")<br>**Value:**<br>Text to be displayed |

| Tag identifier | Description |
|---|---|
| **IMG** | The tag is used to display an image in the specified position. The BMP and PNG image formats are supported.<br><br>**Syntax:**<br>`<IMG xpos = "<X position>" ypos = "<Y position>" name = "<name>" />`<br>`<IMG … xrot="angle X axis" yrot=" angle Y axis " zrot=" angle Z axis " />`<br><br>**Attributes:**<br>• **xpos**<br>X position of the top left corner<br>• **ypos**<br>Y position of the top left corner<br>• **name**<br>Complete path name The path information is made in lowercase letters.<br>• **transparent**<br>Transparent color of the bitmap (see Chapter "Color coding (Page 48)") |
| **IMG** Continued | **Example**:<br>The image is rotated through 34 degrees around the Z axis:<br>`<img xpos = "5" ypos = "23" name = "f:/appl/pic3.bmp" height="355" width="550" zrot="34" color_bk="#ffffff" >`<br>`</img>`<br><br>**Note:**<br>The drive designations vary depending on the system.<br><br><br>**Optional:**<br>If the image display is to differ from the original size, the dimensions can be defined using the attributes **width** and **height**.<br>• **width**<br>Width in pixels<br>• **height**<br>Height in pixels<br><br>**Examples:**<br>`<IMG xpos = "20" ypos = "40" name = "f:/appl/test.bmp" />`<br>`<IMG xpos = "5" ypos = "23" name = "f:/appl/test.bmp" height = "355" width = "550"/>` |

| Tag identifier | Description |
|---|---|
| **IMG** Continued | **Attribute:**<br>**AspectRatioMode**<br>This attribute allows the control of the aspect ratio for an image wherever non-proportional zooming is performed.<br><br>**Values:**<br>• **Ignore**<br>The aspect ratio of the bitmap is adapted to the prescribed height and width.<br>• **Keep** (default)<br>The aspect ratio is retained and ensures that the image is scaled to a rectangle which assumes the greatest expansion within the prescribed height and width.<br>• **KeepByExpanding**<br>The aspect ratio is kept and ensures that the image is scaled to a rectangle which assumes the greatest expansion outside of the prescribed height and width.<br><br>**Example:**<br>`<img xpos = "8" ypos = "30" name = "test.bmp" height="155" width="150" AspectRatioMode="KeepbyExpanding">`<br>`  <property transparent="#00ff00" />`<br>`</img>`<br>`<img xpos = "8" ypos = "200" name = "test.bmp" height="155" width="150" AspectRatioMode="Ignore">`<br>`  <property transparent="#00ff00" />`<br>`</img>`<br>`<img xpos = "200" ypos = "200" name = "test.bmp" height="155" width="150" AspectRatioMode="Keep">`<br>`  <property transparent="#00ff00" />`<br>`</img>`<br><br>• Top - **KeepbyExpanding** property set<br>• Bottom left - **Ignore** property set<br>• Bottom right - **Keep** property set |

| Tag identifier | Description |
|---|---|
| **IMG** Continued | **Attributes:**<br>**ExpandingForRotation**<br>The aspect ration is retained. The image is scaled to a rectangle which corresponds with the bounding box of the rotated and scaled image.<br><br>**Example:**<br>`<img xpos = "8" ypos = "100" name = "pic1.bmp" height="155" width="150" >`<br>`  <property transparent="#ffffff" />`<br>`</img>`<br>`<op>`<br>`  zrot = 45.0;`<br>`</op>`<br>`<img xpos = "205" ypos = "23" name = "pic1.bmp" height="155" width="150" zrot="$zrot" AspectRatioMode="ExpandingForRotation">`<br>`  <property transparent="#ffffff" />`<br>`</img>`<br>`<img xpos = "205" ypos = "200" name = "pic1.bmp" height="155" width="150" zrot="$zrot" >`<br>`  <property transparent="#ffffff" />`<br>`</img>`<br><br>• Left - Image is scaled to a size of 150x155 pixels<br>• Top right - Image is scaled to a size of 150x155 pixels and is rotated around 45 degrees with the **ExpandingForRotation** property<br>• Bottom right - Image is scaled to a size of 150x155 pixels and is rotated around 45 degrees |

| Tag identifier | Description |
|---|---|
| **BOX** | The tag draws a rectangle at the specified position, colored as indicated.<br><br>**Syntax:**<br>`<BOX xpos = "<X position>" ypos = "<Y position>" width = "<X extension>" height = "<Y extension>" color = "<Color code>" />`<br><br>**Attributes:**<br>• **xpos**<br>X position of the top left corner<br>• **ypos**<br>Y position of the top left corner<br>• **width**<br>Extension in X direction (in pixels)<br>• **height**<br>Extension in Y direction (in pixels)<br>• **color**<br>Color coding (see Chapter "Color coding (Page 48)") |
| **FUNCTION** | **Function call**<br>The tag executes the function body, which is specified under the attribute "**name**".<br><br>**Attributes:**<br>• **name** = "Name of the function body"<br>• **return** = "Variable name for saving the result of the function"<br><br>**Values:**<br>List of variables to be transferred to the function body. The variables must be separated by a comma. A maximum of 10 parameters can be transferred.<br>It is also possible to specify constants or text expressions as call parameters. The identifier _T should be placed at the start as a means of identifying text terms.<br><br>**Syntax:**<br>`<FUNCTION name = "<function name>" />`<br>Calling function expects a return value<br>`<FUNCTION name = "<function name>" return = "<Variablenname>" />`<br>Parameter transfer<br>`<FUNCTION name = "<function name>"> var1, var2, var3 </FUNCTION>`<br>`<FUNCTION name = "<function name>"> _T"Text", 1.0, 1 </FUNCTION>`<br><br>**Examples:**<br>See "FUNCTION_BODY" |

| Tag identifier | Description |
|---|---|
| FUNCTION_BODY | **Function body**<br>The tag contains the function body of a subfunction. The function body needs to be programmed within the DialogGui tag.<br><br>**Attributes:**<br>• **name** = "Name of the function body"<br>• **parameter** = "Parameter list" (optional)<br>The attribute lists the transfer parameters that are required. The parameters must be separated by a comma.<br>When the function body is called, the values of the parameters specified in the function call are copied to the transfer parameters listed.<br>• **return** = "true" (optional)<br>If the attribute is set to true then the local variable $return is created. The function's return value which is forwarded to the calling function on quitting the function should be copied to this variable.<br><br>**Syntax:**<br><br>**Function body without parameter**<br>`<FUNCTION_BODY name = "<function name>">`<br>…<br>…<br>…<br>`</ FUNCTION_BODY>`<br><br>**Function body with parameter**<br>`<FUNCTION_BODY name = "<function_name>" parameter = "<p1, p2, p3>">`<br>…<br>`<LET name = "tmp></LET>`<br>`<OP> tmp = p1 </OP>`<br>…<br>`</FUNCTION_BODY>`<br><br>**Function body with return value**<br>`<FUNCTION_BODY name = "<function_name>" parameter = "<p1, p2, p3>" return = "true">`<br>…<br>`<LET name = "tmp></LET>`<br>`<OP> tmp = p1 </OP>`<br>…<br>`<OP> $return = tmp </OP>`<br>`</FUNCTION_BODY>` |

| Tag identifier | Description |
|---|---|
| **FUNCTION_BODY** continued | **Example:**<br>`<function_body name = "test" parameter = "c1,c2,c3" return = "true">`<br>`<LET name = "tmp">0</LET>`<br>`<OP> tmp = c1+c2+c3 </OP>`<br>`<OP> $return = tmp </OP>`<br>`</function_body>`<br>…<br>…<br>…<br>`<LET name = "my_var"> 4 </LET>`<br>`<function name = "test" return = " my_var "> 2, 3,4</function>`<br>`<print text = "result = %d"> my_var </print>`<br>…<br>… |
| **REQUEST** | The tag is used to add a variable to the cyclic reading service (Hotlink). As a consequence, the access time to variables, which are not linked to the control, is reduced.<br>If a function is to be called automatically when a value changes, then the name of the function should be specified as an additional attribute.<br>This tag is only processed within the INIT operation.<br><br>**Attributes:**<br>• **name**<br>Address identifier<br>• **function**<br>Function name<br><br>**Syntax:**<br>`<REQUEST name = "<NC-Variable>" />`<br>or<br><br>`<REQUEST name = "<NC-Variable>" function="<function name>"/>`<br><br>**Example:**<br>`<request name ="plc/mb10" />`<br><br>or<br><br>`<function_body name="my_function" >`<br>`<print text="value changed" />`<br>`</function_body>`<br>…<br>…<br>…<br>`<request name ="plc/mb10" function="my_function"/>` |

| Tag identifier | Description |
|---|---|
| RECALL | This tag can be used in a menu if a navigation is to take place via Recall button. If the tag is programmed, the Recall icon is shown, and the parser may process the Recall button.<br><br>**Syntax:**<br>`<recall>`<br>…<br>…<br>`</recall>` |
| UPDATE_CONTROLS | The tag runs a comparison between the operator controls and the reference variables.<br><br>**Attribute:**<br>• **type**<br>The attribute defines the direction of the data comparison.<br>**= TRUE** – data is read from the reference variables and copied to the operator controls.<br>**= FALSE** – Data is copied from the operator controls to the reference variables.<br><br>**Syntax:**<br>`<UPDATE_CONTROLS type = "<Direction>"/>`<br><br>**Example:**<br>`<SOFTKEY_OK>`<br>`< UPDATE_CONTROLS type="false"/>`<br>`</SOFTKEY_OK>` |

# 1.8 Generating user menus

## 1.8.1 Creating processing cycle forms

The cycle support function allows the automatic creation and decompilation of a cycle call through the form dialog.

To manage this functionality, the following tags are available:

- NC_INSTRUCTION
- CREATE_CYCLE

To mark a cycle form, in the FORM tag, the attribute type should be specified with the value cycle. This marking allows the NC_INSTRUCTION to be processed.

### Example

```
<FORM name = "cycle100_form" type= "CYCLE">
…
…
</FORM>
```

The NC_INSTRUCTION tag contains the cycle call to be generated. All cycle parameters should be reserved using space retainers.

### Example

```
<FORM name = "cycle100_form" type= "CYCLE">
<NC_INSTRUCTION refvar= "cyc_string" >Cycle100 ($p1, $p2, $p3)</
NC_INSTRUCTION>
…
…
…
</FORM
```

The CREATE_CYCLE tag prepares the values saved in the space retainer variables and generates the NC instruction.

This is then copied to the specified variable.

| Tag identifier | Description |
|---|---|
| **NC_INSTRUCTION** | This tag is used to define the NC instruction to be generated.<br>All listed cycle parameters are automatically created as string variables of the **FORM** and are available to the **FORM**.<br><br>Precondition: The FORMattribute **type** is set to the value **CYCLE**.<br><br>**Attribute:**<br>• **refvar**<br>If the tag is assigned a reference variable, all parameters are pre-assigned with the values from the NC block saved in the reference variable.<br>**Syntax:**<br>`<NC_INSTRUCTION> NC instruction with placeholders </NC_INSTRUCTION>`<br><br>**Example:**<br>`<let name="cyc_string" type="string"> Cycle100(0, 1000, 5)</let>`<br>...<br>...<br>...<br>`<FORM name = "cycle100_form" type= "CYCLE">`<br>`<NC_INSTRUCTION refvar= "cyc_string" >Cycle100($p1, $p2, $p3)</ NC_INSTRUCTION>`<br>…<br>…<br>…<br>`</FORM>` |

| Tag identifier | Description |
|---|---|
| CREATE_CYCLE | The tag generates an NC block, whose syntax is defined by the value of the **NC_INSTRUCTION** tag.<br>Before generating the NC instruction, the parser calls the CYCLE_CREATE_EVENT tag of the FORM. This tag can be used to calculate the cycle parameters.<br>**Syntax:**<br>`<CREATE_CYCLE/>`<br><br>**Option:**<br>If a reference variable is specified, the instruction copies the generated call into this variable.<br>`<create_cycle refvar="name" />`<br><br>**Attribute:**<br>• **refvar**<br>If the tag is assigned a reference variable, the NC instruction is copied to this variable.<br><br>**Example:**<br>`<LET name="cyc_string" type="string"> Cycle100(0, 1000, 5)</LET>`<br><br>`<SOFTKEY_OK>`<br>`<caption>OK</caption>`<br>`<CREATE_CYCLE />`<br>`<close_form />`<br>`<navigation>main_menu</navigation>`<br>`</SOFTKEY_OK>`<br><br>`or`<br><br>`<SOFTKEY_OK>`<br>`<caption>OK</caption>`<br>`<CREATE_CYCLE refvar= "cyc_string" />`<br>`<close_form />`<br>`<navigation>main_menu</navigation>`<br>`</SOFTKEY_OK>` |

## 1.8.2 Substitution characters

The system offers the option of defining control properties (attribute values) for the runtime. In order to use this function, the desired property must be set in a local variable and the variable name must be transferred to the tag as an attribute value preceded by the **character $**.

If the tag expects a string as attribute value or value, the $$$ characters must be placed in front of the variable name.

**Example:**

```
<let name="my_ypos">100</let>
<let name="field_name" type="string"></let>

<control name = "edit1" xpos = "322" ypos = "$my_ypos" refvar="nck/
Channel/Parameter/R[1]" />

<op>my_ypos = my_ypos +20 </op>

<control name = "edit2" xpos = "322" ypos = "$my_ypos" refvar="nck/
Channel/Parameter/R[2]" />

<print name ="field_name" text="edit%d">3</print>
<op>my_ypos = my_ypos +20 </op>

<control name = "$field_name" xpos = "322" ypos = "$my_ypos"
refvar="nck/Channel/Parameter/R3]" />

<caption>$$$field_name</caption>
```

# 1.9 Addressing components

Address identifiers for the desired data must be created to address NC variables, PLC blocks or drive data. An address consists of the subpaths **component name** and **variable address**. A slash should be used as a separating character.

## 1.9.1 PLC addressing

Addressing the PLC starts with the path section **plc**.

Table 1-3 The following addresses are permissible:

| | |
|---|---|
| **DBx.DB(f)** | Data block |
| **I(f)x** | Input |
| **Q(f)x** | Output |
| **M(f)x** | Bit memory |
| **V(f)x** | Variable |

| | |
|---|---|
| **DBx.DBXx.b** | Data block |
| **Ix.b** | Input |
| **Qx.b** | Output |
| **Mx.b** | Bit memory |
| **Vx.b** | Variable |

Table 1-4 Data format **f**:

| | |
|---|---|
| B | Byte |
| W | Word |
| D | Double word |

Data format identification is not applicable to bit addressing.

Address **x**:

Valid S7-200 address identifier

Bit addressing:

**b** – Bit number

**Examples:**

```
<data name = "plc/mb170">1</data>
<data name = "i0.1"> 1 </data>
<op> "m19.2" = 1 </op>
```

## 1.9.2 Addressing NC variables

Addressing the NC variables starts with the path section **nck**.

This section is followed by the data address; its structure should be taken from the List Manual NC Variables and Interface Signals.

**Example:**

```
<LET name = "tempStatus"></LET>
<OP> tempStatus ="nck/channel/state/chanstatus" </OP>
```

## 1.9.3 Channel-specific addressing

If no channel number is defined in the address token, access is always to Channel 1 of the operating software.

If it is necessary to read data from a specific channel, the identifier **u** (Unit) with the desired channel number is added to the address.

**Example:**

```
nck/Channel/MachineAxis/actFeedRate[3]
nck/Channel/MachineAxis/actFeedRate[u1, 3]
```

## 1.9.4 Generating NC/PLC addresses during the runtime

It is possible to generate an address identifier during runtime.

In this case, the content of a string variable is used as address in an operation statement as well as in the nc.cap.read and nc.cap.write functions.

Observe the following for this type of addressing mode:

- Write the variable names in quotation marks.
- Use three ‚$' characters as prefix for variable names.

**Syntax:**

```
"$$$variable name"
```

**Example:**

```
<PRINT name="var_adr" text="DB9000.DBW%d"> 2000</PRINT>
<OP> "$$$var_adr" = 1 </OP>
```

## 1.9.5 Addressing drive components

Addressing the drive components starts with the path section **drive**.

Then the drive device is specified:

**CU –** Control Unit

**DC –** Drive Control (Motor Module)

**CULNK** – Expansion Modules (HUBs)

**TM** – Terminal Modules

**LM** – Line Modules

The parameter to be set is added to this section.

**Example:**

```
<LET name="r0002_content"></LET>
<LET name="p107_content"></LET>

<!— Reading of value r0002 on the CU ->

<OP> r0002_content = "drive/cu/r0002" </OP>
<OP> r0002_content = "drive/cu/r0002[CU1]" </OP>

<!— Reading of value r0002 on NX1 ->
<OP> r0002_content = "drive/cu/r0002[CU2]" </OP>

<!— Reading of value p107[0] on the CU ->
<OP> p107_content = "drive/cu/p107[0]" </OP>
<PRINT text="%d"> p107_content </PRINT>

<!— Reading of value p107[0] on the CU ->

<OP> p107_content = "drive/cu/p107[0, CU1]" </OP>
<PRINT text="%d"> p107_content </PRINT>

<!— Reading of value p107[0] on the NX1 ->

<OP> p107_content = "drive/cu/p107[0, CU2]" </OP>
<PRINT text="%d"> p107_content </PRINT>
```

**Addressing the drive objects:**

To address individual objects, the desired object should be entered in square brackets after the parameter.

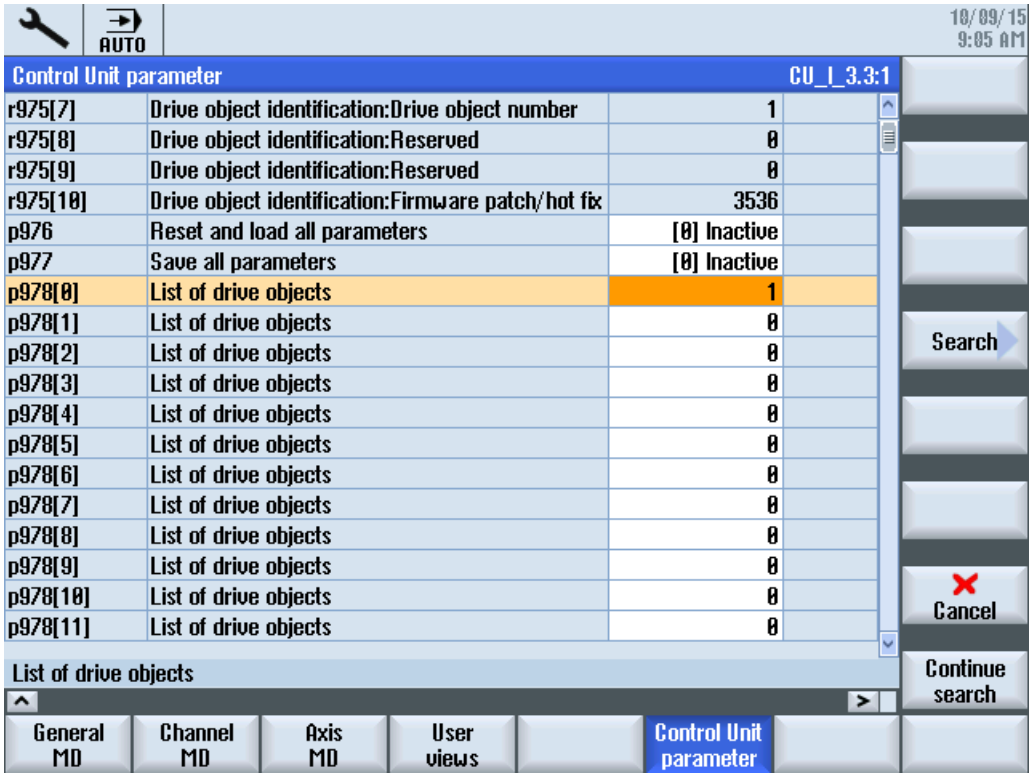



Figure 1-7 Drive parameter p0978 [ ...] at the control

```
Parameter number[do<DO-index>]
```

**Example:**

```
p0092[do1]
```

Alternatively, the drive index can be read from a local variable using **$<variable name>** "substitution characters".

```
z.B. DO$local variable
```

**Example:**

```
<DATA name ="drive/cu/p0092">1</DATA>
<DATA name ="drive/dc/p0092[do1] ">1</DATA>
```

**Indirect addressing:**

```
<LET name = "driveIndex> 0 </LET>
<OP> driveIndex = $ctrlout_module_nr[0, AX1] </OP>
<DATA name ="drive/dc[do$driveIndex]/p0092">1</DATA>
```

## 1.9.6 Example: Determine the DO number for the Motor Module

The DO number of a type 11 (servo) Motor Module can be determined as follows:

All connected drive objects are listed with their slot number in field p978 of the relevant CU. The component type numbers are listed in the field p101 and the component types are listed in the field p107 simultaneously.

A separate indexing is to be used for each of the following component types:

**CU –** Control Units

**DC –** Drive Controls (Motor Module)

**CULNK** – Expansion Modules (HUBs)

**TM** – Terminal Modules

**LM** – Line Modules

The addressing index can be determined by running through the field p107 in ascending order for each connected CU, and the type index is incremented by one each time the desired type occurs. The basic value is one. If NX components are found in this field, the counting on an NX component is not continued until the current array has been run through completely. The NX and CU components are executed in the order found.

### Index determination: CUI with NX

| CUI | | NX1 | | Addressing index | |
|---|---|---|---|---|---|
| | | | | DO | CU |
| p107[0] | [3]SINAMICS | | | | 1 |
| p107[2] | [11]SERVO | | | 1 | |
| p107[3] | [11]SERVO | | | 2 | |
| p107[4] | [11]SERVO | | | 3 | |
| p107[5] | [254]CU-LINK | | | | 2 |
| p107[6] | [11]SERVO | | | 4 | |
| | | p107[1] | [11]SERVO | 5 | |
| | | p107[2] | [11]SERVO | 6 | |
| | | p107[3] | [11]SERVO | 7 | |

This topology contains seven Motor Modules. Indices one to four address the Motor Modules assigned to the CU. Indices five to seven address the Motor Modules of the NX. Index one is to be used to access the CU. The NX is addressed twice by the index.

## Sample scripts: xmldial.xml and drv_sys_hlpfunct.xml

**xmldial.xml**

```
<DialogGui>

 <?include src="f:\appl\drv_sys_hlpfunct.xml" ?>

 <menu name = "main">
  <open_form name = "main_form" />
  <softkey POSITION="1">
   <caption></caption>
   <navigation>main</navigation>
  </softkey>
 </menu>

 <form name="main_form">
  <init>
   <caption>Component arrangement</caption>
    <let name="count" />
    <let name="str" type="string" />
    <let name="do_name" type="string" />
    <let name="cu_name" type="string" />
    <let name="cui_idx" />

    <op>
     cui_idx = 0;
    </op>

    <function name="load_component" />
    <print text="%d CUs found">num_cus</print>

    <function name="calculate_do_index" />

    <control name="list_comp_no_do_idx" xpos="8" ypos="80"
fieldtype="listbox" width="360" height="500" >
     <property item_data="100" />
    </control>

    <op>
     count = 0;
    </op>

    <while>
    <condition>count &lt; 32 && address_idx_map[$count].comp_no != 0</
condition>

    <op>
     do_name= _T"";
    </op>

    <function name="read_do_name_fast" return="do_name">count</function>
    <function name="read_cu_name_fast"
return="cu_name">address_idx_map[$count].cu_idx</function>

    <print name="str" text="%3d %3d %3d %s
%s">address_idx_map[$count].cu_idx, address_idx_map[$count].comp_no,
address_idx_map[$count].do_idx, do_name, cu_name</print>
```

**xmldial.xml**

```
    <function name="control.additem">_T"list_comp_no_do_idx", str, count</
function>

    <op>
     count = count +1;
    </op>
    </while>

   <paint>
    <text xpos="8" ypos="30">Motor moduls</text>
    <text xpos="8" ypos="60">CU</text>
    <text xpos="40" ypos="60">Comp.</text>
    <text xpos="92" ypos="60">DO Index</text>
    <text xpos="172" ypos="60">Name</text>
  </paint>

 </form>
</DialogGui>
```

**drv_sys_hlpfunct.xml**

```
<typedef name="components" type="struct">
 <element name="cu_p978" dim="25" />
 <element name="do_p0101" dim="25" />
 <element name="do_p0107" dim="25" />
</typedef>

<typedef name="comp_no_do_idx_map" type="struct">
 <!-- cu index -->
 <element name="cu_idx" />
 <!-- component number -->
 <element name="comp_no" />
 <!-- address index -->
 <element name="do_idx" />
</typedef>

<let name="componentsList" dim="5" type="components" />
<let name="address_idx_map" dim="32" type="comp_no_do_idx_map" />
<let name="num_cus" />
<let name="_drv_sys_comp_array_size">23</let>

<!-- ------------------------------------------------------------

 function: load_components_description
 This function loads the parameter arrays p978, p101 and p107 into a local
memory

 input:
 cuno: CU index 0 based (index 0 == CUI)

 output:
 componentsList structure

 ------------------------------------------------------------------- -->

<function_body name="load_components_description" parameter="cuno">
 <let name="count" />
 <let name="next_nx" >0</let>
 <let name="cuidx"></let>
 <let name="error" />

 <op>
  count = _drv_sys_comp_array_size;
  next_nx = cuno;
  cuidx = cuno+1;
 </op>

 <print text="gather data" />

 <function name="ncfunc.cap.read" return="error"
rows="$count">componentsList[$cuno].cu_p978, "drive/cu/p0978[0, cu
$cuidx]"</function>
 <function name="ncfunc.cap.read" return="error"
rows="$count">componentsList[$cuno].do_p0101, "drive/cu/p0101[0, cu
$cuidx]"</function>
```

**drv_sys_hlpfunct.xml**

```
 <function name="ncfunc.cap.read" return="error"
rows="$count">componentsList[$cuno].do_p0107, "drive/cu/p0107[0, cu
$cuidx]"</function>

 <print text="gather data finished" />
 <sleep value="20" />

 <op>
  count = 0;
 </op>

 <while>
  <condition>count < _drv_sys_comp_array_size</condition>
  <if>
   <condition>componentsList[$cuno].cu_p978[$count] == 60</condition>
  <then>
   <op>
    next_nx= next_nx +1;
   </op>
  <print text="next nx %d">next_nx</print>
  <function name="load_components_description">next_nx</function>
  </then>
  </if>

 <op>
  count = count+1;
 </op>
 </while>
 <op>
  num_cus = next_nx+1;
 </op>
</function_body>

<!-- ------------------------------------------------------------

 function: calculate_do_index
 This function is looking for components of type 11 (SERVO) and lists these
in the componentsList array.

 input:
 -

 output:
 componentsList array filled

 ------------------------------------------------------------------- -->

<function_body name="calculate_do_index">
 <let name="cuno" />
 <let name="count" />
 <let name="do_index" >1</let>
 <let name="map_index" />
 <while>
  <condition>
   cuno < num_cus</condition>
   <op>
```

**drv_sys_hlpfunct.xml**

```
    count = 0;
   </op>
  <while>
   <condition>count < _drv_sys_comp_array_size</condition>
   <if>
    <condition> componentsList[$cuno].do_p0107[$count] == 11</condition>
    <then>
     <op>
      address_idx_map[$map_index].cu_idx = cuno+1;
      address_idx_map[$map_index].comp_no =
componentsList[$cuno].do_p0101[$count];
      address_idx_map[$map_index].do_idx = do_index;
      do_index = do_index+1;
      map_index = map_index +1;
     </op>
    </then>
   </if>
   <op>
    count = count +1;
   </op>
  </while>
  <op>
   cuno = cuno +1;
  </op>
 </while>
</function_body>
```

## 1.9.7 Addressing machine and setting data

Drive and setting data is identified by the character $ followed by the name of the data.

**Machine data:**

$Mx_<name[**index, AX<axis_number>**]>

**Setting data:**

$Sx_<name[index, AX<axis_number>]>

**x:**

N – General machine or setting data

C – Channel-specific machine or setting data

A – Axis-specific machine or setting data

**Index:**

For a field, the parameter indicates the index of the data.

**AX<axis_number>:**

The required axis (**<axis_number>**) has to be specified for axis-specific data.

Alternatively, the axis index can be read from a local variable using **$<variable name>** "substitution characters".

e.g. AX$localvariable

**Example:**

```
<DATA name ="$MN_AXCONF_MACHAX_NAME_TAB[0] ">X1</DATA>
```

**Direct addressing of the axis:**

```
<DATA name ="$MA_CTRLOUT_MODULE_NR[0, AX1] ">1</DATA>
…
…
```

**Indirect addressing of the axis:**

```
<LET name ="axisIndex"> 1 </LET>
<DATA name ="$MA_CTRLOUT_MODULE_NR[0, AX$axisIndex] ">1</DATA>
```

## 1.9.8 Channel-specific machine data

If no channel number is defined in the address token, access is always to the currently set channel of the operating software.

If it is necessary to read data from a specific channel, the identifier **u** (Unit) with the desired channel number in square brackets is added to the address. In an array-addressing, the channel definition is the last argument in the square brackets.

**Example:**

```
$MC_RESET_MODE_MASK
```

or

```
$MC_AXCONF_GEOAX_ASSIGN_TAB[0]
```

```
$MC_RESET_MODE_MASK[u1]
```

or

```
$MC_AXCONF_GEOAX_ASSIGN_TAB[0, u1]
$MC_RESET_MODE_MASK
```

## 1.9.9 Addressing user data

Addressing user data starts with the path section **gud**, followed by the GUD name.

For a field, after the name, the required field index should be specified in square brackets.

**Example:**
```
<DATA name ="gud/syg_rm[0]"
<OP>"gud/syg_rm[0]" = 10 </op>
```

### Addressing the global user data

Addressing starts with the path section gud, followed by the specification of the area CHANNEL. This address section is followed by the specification of the GUD areas:

| GUD areas | Assignment |
|---|---|
| sgud | Siemens GUD |
| mgud | Machine manufacturer GUD |
| ugud | User GUD |
| GD4 to GD9 | User data extended |

Then enter the GUD name. If an array is to be addressed, the name is followed by the array subscript in square brackets.

**Example:**
```
<data name ="gud/channel/mgud/syg_rm[0]">1</data>
<op>"gud/channel/mgud/syg_rm[0]" = 5*2 </op>
```

### Addressing multi-dimensional arrays

When addressing multi-dimensional arrays, the line index is expected followed by the column index. The indices must be indicated separated by a period.

The following applies to channel-specific GUDs:

- The channel number must be noted with the identifier **u** (unit) followed by the number before or after the line/column specification.
- The sequences must be separated by a comma.
- If no channel number is indicated, access is to the first channel.

**Example:**
```
"gud/Channel/sgud/_WP[u2, 2.0]"
```

or
```
"gud/Channel/sgud/_WP[2.0, u2]"
```

## 1.10 Predefined functions

The script language offers various string processing and standard mathematical functions. The function names listed below are reserved and cannot be overloaded.

| Function name | Description |
|---|---|
| **Ncfunc cap read** | The function copies a value from the specified address into a local variable. If the read operation was error-free, then the return variable contains the value zero.<br><br>Contrary to the operation instruction, in the event of a fault, this function does not interrupt the processing of the script operations.<br><br>**Attributes:**<br>• **return** - Execution status<br>– Value **=** 0 - fault-free<br>– Value **=** 1 - The variables could not be read<br>• **rows** - Number of additional lines of an array to be read (optional)<br>If an array index is defined for the reference variables, as from this index, the function copies the values read into the target variable.<br><br>**Syntax:**<br>`<function name="ncfunc.cap.read" return="error">`<br>`lokale variable, "address"</function>`<br><br>**Example:**<br>`<let name="error"></let>`<br>`<function name="ncfunc.cap.read" return="error"> 3,`<br>`"drive/cu/p0009"</function>`<br>`<if>`<br>`<condition>error != 0</condition>`<br>`<then>`<br>`<break />`<br>`</then>`<br>`</if>`<br><br>or<br><br>`<let name="cu_p978" dim="25" ></let>`<br>`...`<br>`...`<br>`<function name="ncfunc.cap.read" return="error"`<br>`rows="23">cu_p978, "drive/cu/p0978[0, 1]"`<br>`</function>` |

| Function name | Description |
|---|---|
| **Ncfunc cap write** | The function writes a value into the specified variable. If the write operation was error-free, then the return variable contains the value zero.<br><br>Contrary to the operation instruction, in the event of a fault, this function does not interrupt the processing of the script operations.<br><br>**Attributes:**<br>• **return** - Execution status<br>– Value = 0 - faultless<br>– Value = 1 - The variables could not be read<br>• **rows** - Number of additional lines of an array to be written (optional)<br>If an array index is defined for the reference variables, as from this index, the function copies the values into the target variable.<br><br>**Syntax:**<br>`<function name="ncfunc.cap.read" return="error">`<br>`local variable or constant, "address"</function>`<br><br>**Example:**<br>`<let name="error"></let>`<br>`<function name="ncfunc.cap.write" return="error">`<br>`0, "drive/cu/p0009"</function>`<br>`<if>`<br>`<condition>error != 0</condition>`<br>`<then>`<br>`<break />`<br>`</then>`<br>`</if>`<br><br>or<br><br>`<let name="cu_p978" dim="25" ></let>`<br>`...`<br>`...`<br>`<function name="ncfunc.cap.write" return="error"`<br>`rows="23">cu_p978, "drive/cu/p0978[0, 1]"`<br>`</function>` |

| Function name | Description |
|---|---|
| **Ncfunc PI-Service** | Jobs can be transferred to the NCK using the program invocation (PI) service.<br>If the service has been executed error-free, the function returns the value 1 in the return variable.<br><br>**Manipulation of the tool list**<br><br>_N_CREATO - Create tool<br>_N_DELETO - Delete tool<br>_N_CREACE - Create tool cutting edge<br>_N_DELECE - Delete tool cutting edge<br><br>**Activation of work offsets**<br><br>_N_SETUFR - Activates the actual user frame<br>_N_SETUDT - Activates the actual user data<br><br>**Block search**<br><br>_N_FINDBL - Activate block search<br>_N_FINDAB - Cancel block search<br><br>**Syntax:**<br>`<function name="ncfunc.pi_service" return="return var"> pi name, var1, var2, var3, var4, var5 </function>`<br><br>**Attributes:**<br>• name - function name<br>• return- Name of the variable in which the execution result is saved<br>– Value == 1 – job executed successfully<br>– Value == 0 – faulty job<br><br>**Tag values:**<br>• pi name - Name of the PI service (string)<br>• var1 tovar5 - PI specific arguments |

| Function name | Description |
|---|---|
| **Ncfunc PI-Service** Continued | **Arguments:**<br>• _N_CREATO<br>var1 - Tool number<br>• _N_DELETO<br>var1 - Tool number<br>• _N_CREACE<br>var1 - Tool number<br>var2 - Cutting edge number<br>• _N_DELECE<br>var1 - Tool number<br>var2 - Cutting edge number<br>• _N_SETUFR<br>No arguments<br>• _N_SETUDT<br>var1- User data area to be activated<br>– 1 - Tool offset data<br>– 2 - Active basic frame<br>– 3 - Active adjustable frame<br>• _N_FINDBL<br>var1 - Search mode<br>– 2 - Search with contour calculation<br>– 4 - Search for the block end point<br>– 1 - Block search without calculation.<br>• _N_FINDAB<br>No arguments<br><br>**Example:**<br>• Creating a tool – tool number 3<br><br>`<function name="ncfunc.pi_service">_T"_N_CREATO", 3</function>`<br><br>• Delete cutting edge 1 of tool 5<br><br>`<function name="ncfunc.pi_service">_T"_N_DELECE", 5, 1 </function>` |

| Function name | Description |
|---|---|
| **ncfunc chan PI-Service** | The function executes a PI service in a channel-related manner. The channel number is passed after the PI service name. This is followed by all other call parameters.<br><br>**Parameters:**<br>**channel** - Channel number<br><br>**Syntax:**<br>`<function name="ncfunc.chan_pi_service" return="error"> _T"_N_SETUFR", channel, …</function>`<br><br>**Example:**<br>`<let name="chan" >1</let>`<br>`<function name="ncfunc.chan_pi_service" return="error"> _T"_N_SETUFR", chan</function>`<br><br>`<function name="ncfunc.chan_pi_service" return="error"> _T"_N_SETUDT", chan, _T"016", _T"00000", _T"00000"</function>` |
| **Ncfunc display resolution** | This function supplies the conversion rule for floating point numbers defined in the control. A string variable must be provided as variable.<br><br>See also display machine data MD203 DISPLAY_RESOLUTION and MD204 DISPLAY_RESOLUTION_INCH<br><br>**Syntax:**<br>`<function name="ncfunc.displayresolution" return="dislay_res" />`<br><br>**Example:**<br>`<let name="dislay_res" type="string"></let>`<br>`...`<br>`<function name="ncfunc.displayresolution" return="dislay_res" />`<br><br>`<control name = "cdistToGo" xpos = "210" ypos = "156" refvar="nck/Channel/GeometricAxis/ progDistToGo[2]" hotlink="true" height="34" fieldtype="readonly" format="$$$dislay_res" time="superfast" color_bk="#ffffff"/>` |

| Function name | Description |
|---|---|
| **Ncfunc Get drive by axis name** | This function returns the addressing index of a motor module by specifying the associated axis name.<br><br>**Parameter:**<br>**str** - Axis name<br><br>**Return value:**<br>Index of the motor module - Name on a variable in which the determined index is written.<br><br>**Syntax:**<br>`<function name="NCFUNC.GETDRVBYAX_NAME" return="<index>"><axis name></function>`<br><br>**Example:**<br>`<let name="drv_idx" />`<br>`<function name="NCFUNC.GETDRVBYAX_NAME" return="drv_idx">_T"X1"</function>`<br>`<control name = "c_do2" xpos = "140" ypos = "114" width = "320" fieldtype="itemlist" refvar="drive/dc/p10[do$ drv_idx]" hotlink="true" color_bk="#f1f1f1">`<br>`<item value="0">Ready</item>`<br>`<item value="1">Quick commissioning</item>`<br>`<item value="2">Power unit commissioning</item>`<br>`<item value="3">Motor commissioning</item>`<br>`<item value="4">Encoder commissioning</item>`<br>`<item value="5">Technological application/units</item>`<br>`<item value="15">Data sets</item>`<br>`<item value="17">Basic positioning commissioning</item>`<br>`<item value="25">Commissioning the position control</item>`<br>`<item value="29">Download</item>`<br>`<item value="30">Parameter reset</item>`<br>`<item value="95">Safety Integrated commissioning</item>`<br>`</control>`<br>`</init>` |

| Function name | Description |
|---|---|
| **Ncfunc Get drive by axis index** | This function returns the addressing index of a motor module by specifying the associated axis index.<br><br>**Parameter:**<br>**index** - Axis index<br><br>**Return value:**<br>Index of the motor module - Name on a variable in which the determined index is written.<br><br>**Syntax:**<br>`<function name="NCFUNC.GETDRVBYAX_IDX" return="<index>"><axis index></function>`<br><br>**Example:**<br>`<let name="drv_idx" />`<br>`<function name="NCFUNC.GETDRVBYAX_IDX" return="drv_idx">1</function>`<br>`<control name = "c_do2" xpos = "140" ypos = "114" width = "320" fieldtype="itemlist" refvar="drive/dc/p10[`**`do$ drv_idx`**`]" hotlink="true" color_bk="#f1f1f1">`<br>`<item value="0">Ready</item>`<br>`<item value="1">Quick commissioning</item>`<br>`<item value="2">Power unit commissioning</item>`<br>`<item value="3">Motor commissioning</item>`<br>`<item value="4">Encoder commissioning</item>`<br>`<item value="5">Technological application/units</item>`<br>`<item value="15">Data sets</item>`<br>`<item value="17">Basic positioning commissioning</item>`<br>`<item value="25">Commissioning the position control</item>`<br>`<item value="29">Download</item>`<br>`<item value="30">Parameter reset</item>`<br>`<item value="95">Safety Integrated commissioning</item>`<br>`</control>`<br>`</init>` |

| Function name | Description |
|---|---|
| **Ncfunc Get drive by drive name** | This function returns the addressing index of a motor module by specifying the motor module name.<br><br>**Parameter:**<br>**string** - Name of motor module<br><br>**Return value:**<br>Index of the motor module - Name on a variable in which the determined index is written.<br><br>**Syntax:**<br>`<function name="NCFUNC.GETDRVBYDRV_NAME" return="<index>"><drive name></function>`<br><br>**Example:**<br>`<let name="drv_idx" />`<br>`<function name="NCFUNC.GETDRVBYDRV_NAME" return="drv_idx">_T"SERVO_3.13:4"</function>`<br>`<control name = "c_do2" xpos = "140" ypos = "114" width = "320" fieldtype="itemlist" refvar="drive/dc/p10[`**`do$ drv_idx`**`]" hotlink="true" color_bk="#f1f1f1">`<br>`<item value="0">Ready</item>`<br>`<item value="1">Quick commissioning</item>`<br>`<item value="2">Power unit commissioning</item>`<br>`<item value="3">Motor commissioning</item>`<br>`<item value="4">Encoder commissioning</item>`<br>`<item value="5">Technological application/units</item>`<br>`<item value="15">Data sets</item>`<br>`<item value="17">Basic positioning commissioning</item>`<br>`<item value="25">Commissioning the position control</item>`<br>`<item value="29">Download</item>`<br>`<item value="30">Parameter reset</item>`<br>`<item value="95">Safety Integrated commissioning</item>`<br>`</control>`<br>`</init>` |

| Function name | Description |
|---|---|
| **Ncfunc Get drive by bud address** | This function returns the addressing index of a motor module by specifying the bus address of the motor module.<br><br>**Parameter:**<br>**bus** - Bus number<br>**slave** - Slave number<br>**component** - component number<br><br>**Return value:**<br>Index of the motor module - Name on a variable in which the determined index is written.<br><br>**Syntax:**<br>`<function name="NCFUNC.GETDRVBYBUS_ADDR" return="<index>"><bus>,<slave>,<component></function>`<br><br>**Example:**<br>`<let name="drv_idx" />`<br>`<function name="NCFUNC.GETDRVBYBUS_ADDR" return="drv_idx"> 3,13,4 </function>`<br>`<control name = "c_do2" xpos = "140" ypos = "114" width = "320" fieldtype="itemlist" refvar="drive/dc/p10[do$ drv_idx]" hotlink="true" color_bk="#f1f1f1">`<br>`<item value="0">Ready</item>`<br>`<item value="1">Quick commissioning</item>`<br>`<item value="2">Power unit commissioning</item>`<br>`<item value="3">Motor commissioning</item>`<br>`<item value="4">Encoder commissioning</item>`<br>`<item value="5">Technological application/units</item>`<br>`<item value="15">Data sets</item>`<br>`<item value="17">Basic positioning commissioning</item>`<br>`<item value="25">Commissioning the position control</item>`<br>`<item value="29">Download</item>`<br>`<item value="30">Parameter reset</item>`<br>`<item value="95">Safety Integrated commissioning</item>`<br>`</control>`<br>`</init>` |

| Function name | Description |
|---|---|
| **Ncfunc bico to int** | The function converts a string specified in BICO format into an integer value. (see SINAMICS)<br><br>**Syntax:**<br>`<function name="ncfunc.bicotoint" return="integer variable">bico-string</function>`<br><br>**Example:**<br>`<let name="s_np0480_0" type="string"></let>`<br>`<let name="i_p0480_0">0</let>`<br><br>`<function name="ncfunc.bicotoint" return="i_p0480_0">s_np0480_0</function>` |
| **Ncfunc int to bico** | The function converts an integer value into a BICO format string. (see SINAMICS)<br><br>**Syntax:**<br>`<function name="ncfunc.inttobico" return="string variable">integer variable</function>`<br><br>**Example:**<br>`<function name="ncfunc.inttobico" return="s_p0480_0">"drive/dc/ p0480[0, DO2]"</function>` |
| **Ncfunc is bico str valid** | This function returns the value zero for a string specified in BICO format. (see SINAMICS)<br><br>**Syntax:**<br>`<function name="ncfunc.isbicostrvalid" return="integer variable">string varaible</function>`<br><br>**Example:**<br>…<br>`<let name="s_np0480_0" type="string"></let>`<br>…<br>…<br>`<control name = "cp0480_0" xpos = "402" ypos = "76" hotlink="true" refvar="s_np0480_0" >`<br>`<property item_data="4001" />`<br>`</control>`<br>…<br>…<br>`<function name="ncfunc.isbicostrvalid" return="valid">cp0480_0</function>` |

| Function name | Description |
|---|---|
| **Ncfunc password** | This function sets or deletes a password level from the NC.<br>• Set password:<br>The password should be specified for the required password level as parameter.<br>• Delete password:<br>A blank string deletes the password level.<br><br>**Syntax:**<br>`<function name="ncfunc.password">password</function>`<br><br>**Example:**<br>`<let name="password"type="string"></let>`<br><br>`<function name="ncfunc.password" > password</function>`<br>`<function name="ncfunc.password" > _T"CUSTOMER"</function>`<br><br>Delete password:<br>`<function name="ncfunc.password" > _T"" </function>` |
| **Control form color** | This function provides the text or background color of the dialog box as a string (color coding see Chapter "Color coding (Page 48)").<br><br>**Range:**<br>• BACKGROUND – request color value of the background<br>• TEXT – request color value of the text (foreground)<br><br>**Syntax:**<br>`<function name="control.formcolor" return="variable">_T"Area"</function>`<br><br>**Example:**<br>`<let name="bk_color" type="string"></let>`<br><br>`<function name="control.formcolor" return="bk_color">_T"BACKGROUND"</function>` |

| Function name | Description |
|---|---|
| **Control local time** | The function copies the local time in a field with 7 array elements.<br>The name of the variable is expected as call parameter.<br>The following is stored in an array element:<br>• Index 0 - year<br>• Index 1 - month<br>• Index 2 - weekday<br>• Index 3 - day<br>• Index 4 - hour<br>• Index 5 - minute<br>• Index 6 - second<br><br>**Syntax:**<br>`<function name ="control.localtime">_T"time_array"`<br>`</function>`<br><br>**Example:**<br>`<!-- index`<br>`0 = Year`<br>`1 = Month`<br>`2 = Day of week`<br>`3 = Day`<br>`4 = Hour`<br>`5 = Minute`<br>`6 = Second`<br>`-->`<br>`<let name="time_array" dim="7" />`<br><br>`<function name ="control.localtime">_T"time_array"`<br>`</function>` |

| Function name | Description |
|---|---|
| **String to compare** | Two strings are compared with one another from a lexicographical perspective.<br>The function gives a return value of zero if the strings are the same, a value less than zero if the first string is smaller than the second string or a value greater than zero if the second string is smaller then the first string.<br>**Parameter:**<br>**str1** - string<br>**str2** - comparison string<br><br>**Syntax:**<br>`<function name="string.cmp" return ="<int var>" > str1, str2 </function>`<br><br>**Example:**<br>`<let name="rval">0</let>`<br>`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`<br>`<let name="str2" type="string">A brown bear hunts a brown dog.</let>`<br><br>`<function name="string.cmp" return="rval"> str1, str2 </function>`<br>**Result:**<br>rval= 0 |

| Function name | Description |
|---|---|
| **String to compare without making a distinction between uppercase/lowercase** | Two strings are compared from a lexicographical perspective (the comparison is not case-sensitive).<br>The function gives a return value of zero if the strings are the same, a value less than zero if the first string is smaller than the second string or a value greater than zero if the second string is smaller then the first string.<br><br>**Parameter:**<br>**str1** - string<br>**str2** - comparison string<br><br>**Syntax:**<br>`<function name="string.icmp" return ="<int var>" > str1, str2 </function>`<br><br>**Example:**<br>`<let name="rval">0</let>`<br><br>`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`<br>`<let name="str2" type="string">A brown Bear hunts a brown Dog.</let>`<br>`<function name="string.icmp" return="rval"> str1, str2 </function>`<br>**Result:**<br>rval= 0 |
| **String left** | The function extracts the first nCount character from string 1 and copies this to the return variable.<br><br>**Parameter:**<br>**str1** - string<br>**nCount** - number of characters<br><br>**Syntax:**<br>`<function name="string.left" return="<result string>"> str1,  nCount </function>`<br><br>**Example:**<br>`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`<br>`<let name="str2" type="string"></let>`<br><br>`<function name="string.left" return="str2"> str1, 12 </function>`<br>**Result:**<br>str2="A brown bear" |

| Function name | Description |
|---|---|
| **String right** | The function extracts the last nCount character from string 1 and copies this to the return variable.<br><br>**Parameter:**<br>**str1** - string<br>**nCount** - number of characters<br><br>**Syntax:**<br>`<function name="string.right" return="<result string>"> str1, nCount </function>`<br><br>**Example:**<br>`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`<br>`<let name="str2" type="string"></let`<br><br>`<function name="string.right" return="str2"> str1, 10 </function>`<br>**Result:**<br>str2="brown dog." |
| **String middle** | The function extracts the specified number of characters from string 1, starting from the iFirst index, and copies these to the return variable.<br><br>**Parameter:**<br>**str1** - string<br>**iFirst** - start index<br>**nCount** - number of characters<br><br>**Syntax:**<br>`<function name="string.middle" return="<result string>"> str1, iFirst, nCount </function>`<br><br>**Example:**<br>`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`<br>`<let name="str2" type="string"></let`<br><br>`<function name="string.middle" return="str2"> str1, 2, 5 </function>`<br>**Result:**<br>str2="brown" |

| Function name | Description |
|---|---|
| **String length** | The function gives the number of characters in a string.<br><br>**Parameter:**<br>**str1** - string<br><br>**Syntax:**<br>`<function name="string.length" return="<int var>"> str1 </function>`<br><br>**Example:**<br>`<let name="length">0</let>`<br><br>`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`<br>`<function name="string.length" return="length"> str1 </function>`<br>**Result:**<br>length = 31 |
| **Strings to replace** | The function replaces all the substrings found with the new string.<br><br>**Parameter:**<br>**string** - string variable<br>**find string** - string to be replaced<br>**new string** - new string<br><br>**Syntax:**<br>`<function name="string.replace"> string, find string, new string </function>`<br><br>**Example:**<br>`<let name="str1" type="string">A brown bear hunts a brown dog. </let>`<br><br>`<function name="string.replace" > str1, _T"a brown dog" , _T"a big salmon"</function>`<br>**Result:**<br>str1 = "A brown bear hunts a big salmon!" |

| Function name | Description |
|---|---|
| **Strings to remove** | The function removes all the substrings found.<br><br>**Parameter:**<br>**string** - string variable<br>**remove string** - substring to be deleted<br><br>**Syntax:**<br>`<function name="string.remove"> string, remove string </function>`<br><br>**Example:**<br>`<let name="index">0</let>`<br><br>`<let name="str1" type="string">A brown bear hunts a brown dog. </let>`<br>`<function name="string.remove" > str1, _T"a brown dog" </function>`<br>**Result:**<br>str1 = "A brown bear hunts" |
| **Strings to insert** | The function inserts a string at the index specified.<br><br>**Parameter:**<br>**string** - string variable<br>**index** - index (zero based)<br>**insert string** - string to be inserted<br><br>**Syntax:**<br>`<function name="string.insert"> string, index, insert string </function>`<br><br>**Example:**<br>`<let name="str1" type="string">A brown bear hunts. </let>`<br>`<let name="str2" type="string">a brown dog</let>`<br><br>`<function name="string.insert" > str1, 19, str2 </function>`<br>**Result:**<br>str1 = "A brown bear hunts a brown dog" |

| Function name | Description |
|---|---|
| **String delete** | The function deletes the defined number of characters starting from the start position specified.<br><br>**Parameter:**<br>**string** - string variable<br>**start index** - start index (zero based)<br>**nCount** - number of characters to be deleted<br><br>**Syntax:**<br>`<function name="string.delete"> string, start index , nCount </function>`<br><br>**Example:**<br>`<let name="str1" type="string">A brown bear hunts. </let>`<br><br>`<function name="string.delete" > str1, 2, 5 </function>`<br>**Result:**<br>str1 = "A bear hunts" |
| **String find** | The function searches the transferred string for the first match with the substring.<br>If the substring is found, the function provides the index to the first character (starting with zero) or, failing this, -1.<br><br>**Parameter:**<br>**string** - string variable<br>**find string** - string to be found<br>**startindex** – start index (optional)<br><br>**Syntax:**<br>`<function name="string.find" return="<int val>"> str1, find string </function>`<br><br>**Example:**<br>`<let name="index">0</let>`<br>`<let name="str1" type="string">A brown bear hunts a brown dog. </let>`<br>`<function name="string.find" return="index"> str1, _T"brown" </function>`<br>**Result:**<br>Index = 2<br>**or**<br>`<function name="string.find" return="index"> str1, _T"brown", 1 </function>` |

| Function name | Description |
|---|---|
| **String reverse find** | The function searches the transferred string for the last match with the substring.<br>If the substring is found, the function provides the index to the first character (starting with zero) or, failing this, -1.<br><br>**Parameter:**<br>**string** - string variable<br>**find string** - string to be found<br>**startindex** – start index (optional)<br><br>**Syntax:**<br>`<function name="string.reversefind" return="<int val>"> str1, find string </function>`<br><br>**Example:**<br>`<let name="index">0</let>`<br>`<let name="str1" type="string">A brown bear hunts a brown dog. </let>`<br>`<function name="string.reversefind" return="index"> str1, _T"brown" </function>`<br>**Result:**<br>Index = 21<br>**or**<br>`<function name="string.reversefind" return="index"> str1, _T"brown", 10 </function>`<br>**Result:**<br>Index = 2 |
| **String trim left** | The function trims the starting characters from a string.<br><br>**Parameter:**<br>**str1** - string variable<br><br>**Syntax:**<br>`<function name="string.trimleft" > str1 </function>`<br><br>**Example:**<br>`<let name="str1" type="string">test trim left</let>`<br>`<function name="string.trimleft" > str1 </function>`<br>**Result:**<br>str1 = "test trim left" |

| Function name | Description |
|---|---|
| **String trim right** | The function trims the closing characters from a string.<br><br>**Parameter:**<br>**str1** - string variable<br><br>**Syntax:**<br>`<function name="string.trimright" > str1 </function>`<br><br>**Example:**<br>`<let name="str1" type="string"> test trim right`<br>`</let>`<br>`<function name="string.trimright" > str1`<br>`</function>`<br>**Result:**<br>str1 = "test trim right" |
| **Sine** | The function calculates the sine of the value transferred in degrees.<br><br>**Parameter:**<br>**double** - angle<br><br>**Syntax:**<br>`<function name="sin" return="<double val>"> double`<br>`</function>`<br><br>**Example:**<br>`<let name= "sin_val" type="double"></let>`<br>`<function name="sin" return="sin_val"> 20.0`<br>`</function>` |
| **Cosine** | The function calculates the cosine of the value transferred in degrees.<br><br>**Parameter:**<br>**double** - angle<br><br>**Syntax:**<br>`<function name="cos" return="<double val>"> double`<br>`</function>`<br><br>**Example:**<br>`<let name= "cos_val" type="double"></let>`<br>`<function name="cos" return="cos_val"> 20.0`<br>`</function>` |

| Function name | Description |
|---|---|
| **Tangent** | The function calculates the tangent of the value transferred in degrees.<br><br>**Parameter:**<br>**double** - angle<br><br>**Syntax:**<br>`<function name="tan" return="<double val>"> double </function>`<br><br>**Example:**<br>`<let name= "tan_val" type="double"></let>`<br>`<function name="tan" return="tan_val"> 20.0 </function>` |
| **ARCSIN** | The function calculates the arcsine of the value transferred in degrees.<br><br>**Parameter:**<br>**double** - x in the range from -PI/2 to +PI/2<br><br>**Syntax:**<br>`<function name="arcsin" return="<double val>"> double </function>`<br><br>**Example:**<br>`<let name= "arcsin_val" type="double"></let>`<br>`<function name="arcsin" return="arcsin_val"> 20.0 </function>` |
| **ARCOS** | The function calculates the arccosine of the value transferred in degrees.<br><br>**Parameter:**<br>**double -** x in the range from -PI/2 to +PI/2<br><br>**Syntax:**<br>`<function name="arcos" return="<double val>"> double </function>`<br><br>**Example:**<br>`<let name= "arccos_val" type="double"></let>`<br>`<function name="arccos" return="arccos_val"> 20.0 </function>` |

| Function name | Description |
|---|---|
| **ARCTAN** | The function calculates the arctan of the value transferred in degrees.<br><br>**Parameter:**<br>**double** - arctan of y/x<br><br>**Syntax:**<br>`<function name="arctan" return="<double val>"> double </function>`<br><br>**Example:**<br>`<let name= "arctan_val" type="double"></let> <function name="arctan" return="arctan_val"> 20.0 </function>` |
| File processing | |

| Function name | Description |
|---|---|
| **Reading a file** | The function reads the contents of the specified file into a string variable.<br>The number of characters to be read can optionally be specified as a second parameter.<br><br>**Attribute:**<br>**name -** The file name should be written in lowercase letters.<br>Files in other directories are accessed via a relative path that uses the appl or dvm directory as a starting point.<br>**return** - name of the local variable<br><br>**Parameter:**<br>**progname** - file name<br>**number of characters** - number of characters to be read in bytes (optional):<br><br>**Syntax:**<br>`<function name="doc.readfromfile" return="<string var>"> progname, number of characters </function>`<br><br>**Example:**<br>`<let name = "my_var" type="string" ></let>`<br><br>NC file system<br>`<function name="doc.readfromfile" return="my_var">_T"n:\mpf\test.mpf" </function>`<br><br>CompactFlash card<br>`<function name="doc.readfromfile" return="my_var">_T"f:\appl\test.mpf" </function>`<br><br>or<br><br>`<function name="doc.readfromfile" return="my_var">_T".\test.mpf" </function>` |

| Function name | Description |
|---|---|
| **Writing to a file** | The function writes the contents of a string variable to the file specified.<br>The file name should be written in lowercase letters.<br>Files in other directories are accessed via a relative path that uses the appl or dvm directory as a starting point.<br><br>**Parameter:**<br>**progname** - file name<br>**str1** - string<br><br>**Syntax:**<br>`<function name="doc.writetofile" > progname, str1 </function>`<br><br>**Example:**<br>`<let name = "my_var" type="string" > file content </let>`<br><br>NC file system<br>`<function name="doc.writetofile">_T"n:\mpf\test.mpf", my_var </function>`<br><br>CompactFlash card<br>`<function name="doc.writetofile">_T"f:\appl\test.mpf", my_var </function>`<br><br>or<br><br>`<function name="doc.writetofile">_T".\test.mpf", my_var </function>` |

| Function name | Description |
|---|---|
| **Deleting a file** | The function removes the file specified from the directory.<br>The file name should be written in lowercase letters.<br>Files in other directories are accessed via a relative path that uses the appl or dvm directory as a starting point.<br><br>**Parameter:**<br>**progname** - file name<br><br>**Syntax:**<br>`<function name="doc.remove" > progname </function>`<br><br>**Example:**<br>NC file system<br>`<function name="doc.remove">_T"n:\mpf\test.mpf" </function>`<br><br>CompactFlash card<br>`<function name="doc.remove">_T"f:\appl\test.mpf" </function>`<br><br>or<br><br>`<function name="doc.remove">_T".\test.mpf" </function>` |

| Function name | Description |
|---|---|
| **Extracting script parts** | The function copies a dialog description embedded in a part program into the specified local variable.<br><br>The call parameters to be specified are the program name, the dialog name, and a variable for storing the main menu name. If the name of the dialog description was found in the part program, the return variable contains this description. If the content of the variable is stored in a file, the script can be executed with an indirect call.<br><br>The system provides a script that extracts the dialog description from the active part program and activates the dialog. This script can be called in an MMC command to activate the screen associated with the part program.<br><br>**Syntax:**<br>`<function name="doc.loadscript" return="<name of script variable>">progname, _T"dialog part name", main menu </function>`<br><br>**Attribute**:<br>**return** - variable in which the extracted script is stored<br><br>**Parameters:**<br>**progname** - full path to the program. (The path name can be passed to the function in DOS notation.)<br>**main menu** - the menu name found is copied into this variable<br>**dialog part name** - tag name in which the dialog description is embedded<br><br>**Example:**<br>`<function name="doc.loadscript" return="contents">prog_name, _T"main_dialog", entry</function>` |

| Function name | Description |
|---|---|
| **Exist** | If the file exists, the function returns the value 1.<br>The file name should be written in lowercase letters.<br>Files in other directories are accessed via a relative path that uses the appl or dvm directory as a starting point.<br><br>**Parameter:**<br>**progname** - file name<br><br>**Syntax:**<br>`<function name="doc.exist" return="<int_var>" > progname </function>`<br><br>**Example:**<br>`<let name ="exist">0</let>`<br><br>NC file system<br>`<function name="doc.exist" return="exist">_T"n:\mpf \test.mpf" </function>`<br><br>CompactFlash card<br>`<function name="doc.exist" return="exist">_T"f:\appl \test.mpf" </function>`<br><br>or<br><br>`<function name="doc.exist" return="exist">_T". \test.mpf" </function>` |
| **NC program selection** | The function selects the program specified for execution. The program must be stored in the NC file system.<br><br>**Parameter:**<br>**progname** - file name<br><br>**Syntax:**<br>`<function name="ncfunc.select"> progname </function>`<br><br>**Example:**<br>NC file system<br>`<function name="ncfunc.select"> _T"n:\mpf\test.mpf" </function>` |

| Function name | Description |
|---|---|
| **Setting an individual bit** | The function is used to manipulate individual bits of the specified variables.<br>The bits can either be set or reset.<br><br>**Syntax:**<br>`<function name="ncfunc.bitset" refvar="address" value="set/reset" > bit0, bit1, … bit9 </function>`<br><br>**Attributes:**<br>**refvar** - specifies the name of the variable, in which the bit combination should be written<br>**value** – bit value, value range 0 and 1<br><br>**Values:**<br>The bit numbers starting with zero should be transferred as function values.<br>A maximum of 10 bits per call can be modified.<br><br>**Example:**<br>`<function name="ncfunc.bitset" refvar="nck/Channel/Parameter/R[1]" value="1" > 0, 2, 3, 7 </function>`<br><br>`<function name="ncfunc.bitset" refvar="nck/Channel/Parameter/R[1]" value="0" > 1, 4 </function>` |
| **Delete control** | The function deletes the specified picture control.<br><br>**Syntax:**<br>`<function name="control.delete"> control name </function>`<br><br>**Attribute:**<br>**name** – function name<br><br>**Value:**<br>**control name** – name of the control<br><br>**Example:**<br>`<function name="control.delete"> _T"my_editfield" </function>` |

| Function name | Description |
|---|---|
| **Add Item** | The function inserts a new element at the end of the list.<br>Note:<br>The function is only available for the control types "listbox" and "graphicbox".<br><br>**Syntax:**<br>`<function name="control.additem"> control name, item </function>`<br><br>**Attribute:**<br>**name** – function name<br><br>**Values:**<br>**control name** – control name<br>**item** - expression to be inserted<br>**itemdata** - integer value; defined by the user<br><br>**Example:**<br>`<let name ="itemdata">1</let>`<br>…<br>…<br>…<br>`<op> item_string = _T"text1" </op>`<br>`<function name="control.additem">_T"listbox1", item_string, itemdata`<br>`</function>` |

| Function name | Description |
|---|---|
| **Insert Item** | The function inserts a new element at the specified position.<br>Note:<br>The function is only available for the control type "listbox".<br><br>**Syntax:**<br>`<function name="control.insertitem"> control name, index, item, itemdata </function>`<br><br>**Attribute:**<br>**name** – function name<br><br>**Values:**<br>**control name** - Control name<br>**index** - Position starting with zero<br>**item** - expression to be inserted<br>**itemdata** - integer value; defined by the user<br><br>**Example:**<br>`<let name ="itemdata">1</let>`<br>…<br>…<br>…<br>`<op> item_string = _T"text2" </op>`<br>`<function name="control.insertitem">_T"listbox1", 1, item_string, itemdata </function>` |
| **Delete item** | The function deletes an element at the specified position.<br>Note:<br>The function is only available for the control type "listbox".<br><br>**Syntax:**<br>`<function name="control.deleteitem"> control name, index </function>`<br><br>**Attribute:**<br>**name** - Function name<br><br>**Values:**<br>**control name** - Control name<br>**index**– index starting at 0<br><br>**Example:**<br>`<function name="control.deleteitem">_T"listbox1", 1</function>` |

| Function name | Description |
|---|---|
| **Load Item** | The function inserts a list of expressions into the control.<br>The function is only available for the control types "listbox" and "graphicbox".<br><br>**Syntax:**<br>`<function name="control.loaditem"> control name, list </function>`<br><br>**Attribute:**<br>**name** – function name<br><br>**Values:**<br>**control name** – control name<br>**list**- string variable<br><br>**Structure of the list:**<br>The list contains a number of expressions, which must be separated from one another using a \n.<br><br>**Example:**<br>`<let name="item_string" type="string"></let>`<br>`<let name="plotlist" type="string"></let>`<br>…<br>…<br>`<print name ="item_string" text="p; %f; %f; %f; %f\ \n">s_z, s_x</print>`<br>`<op>plotlist = plotlist + item_string</op>`<br>`<print name ="item_string" text="l; %f; %f; %f; %f\\n">s_z, s_x, e_z, e_x </print>`<br>`<op>plotlist = plotlist + item_string</op>`<br>`<op> s_x = e_x </op>`<br>`<op> s_z = e_z</op>`<br>`<op> e_x = s_x + 10 </op>`<br>`<op> e_z = s_z - 100 </op>`<br>`<print name ="item_string" text="l; %f; %f; %f; %f\\n">s_z, s_x, e_z, e_x </print>`<br>`<op>plotlist = plotlist + item_string</op>`<br><br>`<function name="control.loaditem">_T"gbox", plotlist</function>` |

| Function name | Description |
|---|---|
| **Empty** | The function deletes the contents of the specified list box or graphic box controls.<br><br>**Syntax:**<br>`<function name="control.empty"> control name, </function>`<br><br>**Attribute:**<br>**name** – function name<br><br>**Values:**<br>**control name** – control name<br><br>**Example:**<br>`<function name="control.empty">_T"listbox1" </function>` |
| **Get focus** | The function supplies the name of the control, which has the input focus.<br><br>**Syntax:**<br>`<function name="control.getfocus" return="focus_name" />`<br><br>**Attributes:**<br>**name** – function name<br>**return** – a string variable should be specified, into which the control name is copied.<br><br>**Example:**<br>`<let name>="focus_field" type="string"></let>`<br>`<function name="control.getfocus" return="focus_field"/>` |

| Function name | Description |
|---|---|
| **Set focus** | The function sets the input focus to the specified control.<br>The Controlname should be transferred as text expression of the function.<br><br>**Syntax:**<br>`<function name="control.setfocus"> control name </function>`<br><br>**Attribute:**<br>**name** – function name<br><br>**Value:**<br>**control name** – name of the control<br><br>**Example:**<br>`<function name="control.setfocus" ">_T"listbox1" </function>` |
| **Get cursor selection** | For a list box, the function supplies the cursor index.<br>The Controlname should be transferred as text expression of the function.<br><br>**Syntax:**<br>`<function name="control.getcurssel" return="var">control name </function>`<br><br>**Example:**<br>`<let name>="index"></let>`<br>`<function name="control.getcurssel" return="index">_T"listbox1"</function>` |
| **Set cursor selection** | For a list box, the function sets the cursor to the appropriate line.<br>The Controlname should be transferred as text expression of the function.<br><br>**Syntax:**<br>`<function name="control.setcurssel" > control name, index</function>`<br><br>**Example:**<br>`<let name>="index">2</let>`<br>`<function name="control.setcurssel" ">_T"listbox1",index</function>` |

| Function name | Description |
|---|---|
| **Get Item** | For a list box, the function copies the contents of the selected line to the specified variable.<br>A string variable should be specified as reference variable.<br>The Controlname should be transferred as text expression of the function.<br><br>**Syntax:**<br>`<function name="control.getitem" return="var">`<br>`control name, index </function>`<br><br>**Example:**<br>`<let name>="index">2</let>`<br>`<let name>="item" type="string"></let>`<br><br>`<function name="control.getitem" return="item"`<br>`">_T"listbox1",index</function>` |
| **Get Item Data** | For a list box, the function copies the user-specific allocated value of an element to the specified variable.<br>For an edit control, the function copies the user-specific allocated value (item_data) to the specified variable.<br>An integer variable should be specified as reference variable.<br>The Controlname should be transferred as text expression of the function.<br><br>**Syntax:**<br>`<function name="control.getitemdata" return="var">`<br>`control name, index </function>`<br><br>**Example:**<br>`<let name>="index">2</let>`<br>`<let name>="itemdata"></let>`<br><br>`<function name="control.getitemdata"`<br>`return="itemdata" ">_T"listbox1",index</function>` |

| Function name | Description |
|---|---|
| **Image Box Set** | This function is used to control the visible area of the image box control. The call parameters to be specified are the control name, the control command, and the associated values.<br><br>**Syntax:**<br>`<function name="control.imageboxset">control name, command, command parameter</function>`<br><br>**Call parameters:**<br>• **x_offs**<br>The function shifts the image portion to the specified X-position A further parameter to be specified is the coordinate of the left corner.<br>• **y_offs**<br>The function shifts the image portion to the specified Y-position A further parameter to be specified is the coordinate of the upper corner.<br>• **xy_offs**<br>The function shifts the image portion to the specified X/Y-positions Further parameters to be specified are the coordinates of the left and upper corners.<br>• **followCursor**<br>The image portion follows the set cursor position.<br>• **zoomplus**<br>The image is enlarged by a factor of 0.12.<br>• **zoomminus**<br>The image is reduced by a factor of 0.12.<br>• **autozoom**<br>The image will automatically be scaled to fit the display area.<br>• **Update**<br>The control is redrawn.<br>• **SetBkColor**<br>The command sets the specified background color. A further parameter to be specified is the color coding.<br>• **SetCursorRect**<br>The portion specified as a rectangle is shifted into the visible area.<br>• **SetAnimationState(applies to GIF animations only)**<br>– **start** - The command starts an animation.<br>– **stop** - The command stops an animation.<br><br>**Example:**<br>`<function name="control.imageboxset">_T"topoview", _T"zoomplus"</function>` |

| Function name | Description |
|---|---|
| **Image Box Get** | This function is for querying the control properties image box The call parameters to be specified are the control command and the associated values.<br><br>**Syntax:**<br>`<function name="control.imageboxget">control name, command, command parameter</function>`<br><br>**Call parameters:**<br>**GetViewSize**<br>Returns the size of the display area<br>A further parameter to be specified is the variable with structure type SIZE.<br><br>**Example:**<br>`<let name="view_size" type="size" />`<br>`<function name="control.imageboxget">_T"topoview", _T"GetViewSize", view_size</function>` |
| **Bitmap Dim** | The function copies the dimension of a bitmap back into a variable with structure type SIZE. To define the type, file struct_def.xml must be included in the project.<br><br>**Syntax:**<br>`<function name="bitmap.dim" >name, variable type </function>`<br><br>**Parameters:**<br>**name** - file path<br>**variable type** - variable name of a variable of type SIZE<br><br>**Example:**<br>`<let name="bmp_size" type="size" />`<br><br>`<function name="bitmap.dim" >_T"test.bmp", bmp_size </function>` |
| **Screen Resolution** | The function copies the absolute screen resolution of the system back into a variable with structure type SIZE. To define the type, file struct_def.xml must be included in the project.<br><br>**Syntax:**<br>`<function name="hmi.screen_resolution" >varialble type </function>` |
| **Get HMI Resolution** | The function copies the screen resolution used by SINUMERIK Operate back into a variable with structure type SIZE. To define the type, file struct_def.xml must be included in the project.<br><br>**Syntax:**<br>`<function name="hmi.get_hmi_resolution" >variable type </function>` |

| Function name | Description |
|---|---|
| **Get Caption Height** | The function returns the title bar height in pixels.<br><br>**Syntax:**<br>`<function name="hmi.get_caption_heigt" return="<return var>" />`<br><br>**Attributes:**<br>**return** - integer variable |
| **Abs** | This function returns the absolute value of the specified number.<br><br>**Syntax:**<br>`<function name="abs" return="var"> value </function>` |
| **SDEG** | The function converts the specified value into degrees.<br><br>**Syntax:**<br>`<function name="sdeg" return="var"> value </function>` |
| **SRAD** | The function converts the specified value into RADian.<br><br>**Syntax:**<br>`<function name="srad" return="var"> value </function>` |
| **SQRT** | The function calculates the square root of the specified value.<br><br>**Syntax:**<br>`<function name="sqrt" return="var"> value </function>` |
| **ROUND** | The function rounds of the transferred number to the specified number of decimal places. If the number of decimal places is not specified, then the function rounds off the number, taking into account the first decimal place.<br><br>**Syntax:**<br>`<function name="round" return="var"> value, nDecimalPlaces </function>` |
| **FLOOR** | The function supplies the largest possible integer value, which is less than or equal to the transferred value.<br><br>**Syntax:**<br>`<function name="floor" return="var"> value </function>` |
| **CEIL** | The function supplies the smallest possible integer value, which is greater than or equal to the transferred value.<br><br>**Syntax:**<br>`<function name="ceil" return="var"> value </function>` |

| Function name | Description |
|---|---|
| **LOG** | The function calculates the logarithm of the specified value.<br><br>**Syntax:**<br>`<function name="log" return="var"> value </function>` |
| **LOG10** | The function calculates the common (decadic) logarithm of the specified value.<br><br>**Syntax:**<br>`<function name="log10" return="var"> value </function>` |
| **POW** | The function calculates the value "$a^b$".<br><br>**Syntax:**<br>`<function name="pow" return="var"> a, b </function>` |
| **MIN** | The function compares the transferred value and returns the lower of the values.<br><br>**Syntax:**<br>`<function name="min" return="var"> value1, value2 </function>` |
| **MAX** | The function compares the transferred value and returns the higher of the values.<br><br>**Syntax:**<br>`<function name="max" return="var"> value1, value2 </function>` |
| **RANDOM** | The function returns a pseudo random number.<br><br>**Syntax:**<br>`<function name="random" return="var" </function>` |

| Function name | Description |
|---|---|
| **MMC** | **Function**:<br>You can use the MMC command to display user-defined dialog boxes (dialog screens) from the part program in SINUMERIK Operate. The appearance of the dialog boxes is defined in a pure text configuration (XML file in the manufacturer's folder). The system software remains unchanged.<br>Due to changes in the operate base system, the parameters are as follows:<br>XML → CYCLES or POPUPDLG<br>XML_ON → PICTURE_ON<br>XML_OFF → PICTURE_OFF<br>**Syntax:**<br>MMC ("operating area, command, XML script file, menu name, reserved, reserved, display time or acknowledgment variable, reserved", "acknowledgment mode")<br>**Meaning:**<br>• MMC<br>Calling the dialog box interactively from the part program in SINUMERIK Operate.<br>• EasyXml<br>Identification for user dialogs that are to be started from a part program.<br>• XML_ON or XML_OFF<br>Command: Display selection or display deselection.<br>• XML file<br>Name of the XML script file<br>• Menu<br>Name of the menu tag that manages the dialog to be displayed<br>• reserved<br>reserved for SINUMERIK Operate<br>• reserved<br>reserved for SINUMERIK Operate<br>• Time<br>Display time of the dialog for acknowledgment mode "N"<br>• reserved<br>reserved for SINUMERIK Operate<br>• Acknowledgment mode<br>"S" synchronous, acknowledgment via the "OK" softkey<br>"N" asynchronous, dialog closes after the set time |

| Function name | Description |
|---|---|
| **MMC** continued | **Example of synchronous call:**<br>Due to changes in the operate base system, the parameters are as follows:<br>XML → CYCLES or POPUPDLG<br>XML_ON → PICTURE_ON<br>XML_OFF → PICTURE_OFF<br>NC instruction<br>`MMC("EASYXML oder CYCLES",XML_ON,mmc_cmd.xml,cmd1,,,,,","S")`<br>`File: mmc_cmd.xml`<br><br>`<menu name = "cmd1">`<br>`<open_form name = "cmd1_form" />`<br>`<softkey_ok>`<br>`<close_form />`<br>`</softkey_ok>`<br>`</menu>`<br><br>`<form name = "cmd1_form" xpos ="12" ypos="100" width="500" height="240">`<br>`<init>`<br>`</init>`<br>`<paint>`<br>`</paint>`<br>`</form>`<br><br>**Example of asynchronous call (acknowledgment not expected):**<br>Due to changes in the operate base system, the parameters are as follows:<br>XML → CYCLES or POPUPDLG<br>XML_ON → PICTURE_ON<br>XML_OFF → PICTURE_OFF<br>NC instruction<br>`MMC("EASYXML oder CYCLES",PICTURE_ON,mmc_cmd.xml,cmd1,,,10,,","N")`<br><br>`File: mmc_cmd.xml <menu name = "cmd1">`<br><br>`<open_form name = "cmd1_form" />  <softkey_ok>`<br>`<close_form />  </softkey_ok>`<br>`</menu>`<br><br>`<form name = "cmd1_form" xpos ="12" ypos="100" width="500" height="240">`<br>`<init>`<br>`</init>`<br>`<paint>`<br>`</paint>`<br>`</form>` |

| Function name | Description |
|---|---|
| **MMC** continued | **Example of extraction of script parts from a part program:**<br>Due to changes in the operate base system, the parameters are as follows:<br>XML → CYCLES or POPUPDLG<br>XML_ON → PICTURE_ON<br>XML_OFF → PICTURE_OFF<br>NC instruction<br>`MMC("XML,XML_ON,XMLDIAL_EMB.XML,main","S")`<br>`File: xmldial_emb.xml`<br><br>`<DialogGui>`<br>` <let name="menu_name" type="string">main</let>`<br>` <let name="script_loaded">0</let>`<br>` <menu name = "main">`<br>` <if>`<br>` <condition>script_loaded == 0</condition>`<br>`  <then>`<br>` <function name ="load_current_program" />`<br>` <dynamic_include src="__tmp.xml" />`<br>`  <op>`<br>`  script_loaded = 1;`<br>`  </op>`<br>` <navigation>$$$menu_name</navigation>`<br>` </then>`<br>` </if>`<br>` <softkey_back>`<br>` <close_form />`<br>` </softkey_back>`<br>` </menu>`<br>` <function_body name ="load_current_program" >`<br>` <let name="prog_name" type= "string"/>`<br>` <let name="contents" type= "string"/>`<br>` <let name="entry" type= "string"/>`<br>` <let name="len" />`<br>` <op> prog_name = "nck/Channel/ProgramInfo/workPandProgName"</op>`<br>` <function name="DOC.LOADSCRIPT" return="contents">prog_name, _T"main_dialog", entry</function>`<br>` <function name="string.length" return="len">contents</function>`<br>` <if>`<br>` <condition>len > 0</condition>`<br>` <then>`<br>`  <op>`<br>`  menu_name = entry;`<br>`  </op>`<br>` <function name="doc.writetofile" >_T"__tmp.xml", contents</function>`<br>` </then>`<br>` </if>`<br>` </function_body>`<br>`</DialogGui>` |

| Function name | Description |
|---|---|
| **MMC** continued | Programming example<br><br>;<main_dialog entry="rpara_main"><br>; <let name="xpos" /><br>; <let name="ypos" /><br>; <let name="field_name" type="string" /><br>; <let name="num" /><br>; <menu name="rpara_main"><br>; <open_form name="rpara_form"/><br>; <softkey_back><br>; <close_form /><br>; </softkey_back><br>;</menu><br>;<br>;<form name="rpara_form"><br>; <init><br>; <caption>test mask</caption><br>; <let name="count" >0</let><br>; <op><br>; xpos = 120;<br>; ypos = 34;<br>;<br>; "nck/Channel/Parameter/R[10]" = 10;<br>; </op><br>; <!-- load the number of controls --><br>; <op><br>; num = "nck/Channel/Parameter/R[10]";<br>; </op><br>;<br>; <while><br>; <condition>count < num</condition><br>; <print name="field_name" text="edit%d">count</print><br>;<br>; <control name = "$field_name" xpos = "$xpos" ypos = "$ypos" refvar="nck/Channel/Parameter/R[$count]" hotlink="true" /><br>; <op><br>; ypos = ypos +24;<br>; count = count +1;<br>; </op><br>; </while><br>;<br>; </init><br>;<br>; <paint><br>; <op><br>; xpos = 8;<br>; ypos = 36;<br>; count = 0;<br>; </op><br>; <while><br>; <condition>count < num</condition><br>; <print name="field_name" text="R-Parameter %d">count </print> |

| Function name | Description |
|---|---|
| | ;<br>; <text xpos = "$xpos" ypos = "$ypos" >$$<br>$field_name</text><br>; <op><br>; ypos = ypos +24;<br>; count = count +1;<br>; </op><br>; </while><br>;<br>; </paint><br>;<br>;</form><br>;<br>;</main_dialog><br>…<br>G94 F100<br>MMC(“XML,XML_ON,XMLDIAL_EMB.XML,main","A")<br>…<br>…<br>…<br>MMC(“XML,XML_OFF,XMLDIAL_EMB.XML,main","A")<br>G4 F2<br>M2 |

| Function name | Description |
|---|---|
| **String find** | The function searches the transferred string for the first match with the substring.<br>If the substring is found, the function returns the index to the first character (starting with zero) or, if not, -1.<br><br>**Parameter:**<br>**string** - string variable<br>**findstring** - string to be found<br>**startindex** – start index (optional)<br><br>**Syntax:**<br>`<function name="string.find" return="<int val>"> str1, find string </function>`<br><br>**Example:**<br>`<let name="index">0</let> <let name="str1" type="string">A brown bear hunts a brown dog. </let> <function name="string.find" return="index"> str1, _T"brown" </function>`<br><br>**Result:**<br>Index = 2<br><br>**or**<br>`<function name="string.find" return="index"> str1, _T"brown", 1 </function>` |

| Function name | Description |
|---|---|
| **String reverse find** | The function searches the transferred string for the last match with the substring.<br>If the substring is found, the function returns the index to the first character (starting with zero) or, if not, -1.<br><br>**Parameter:**<br>**string** - string variable<br>**find string** - string to be found<br>**startindex** – start index (optional)<br><br>**Syntax:**<br>`<function name="string.reversefind" return="<int val>"> str1, find string </function>`<br><br>**Example:**<br>`<let name="index">0</let>`<br>`<let name="str1" type="string">A brown bear hunts a brown dog. </let>`<br>`<function name="string.reversefind" return="index"> str1, _T"brown" </function>`<br><br>**Result:**<br>Index = 21<br><br>**or**<br>`<function name="string.reversefind" return="index"> str1, _T"brown" 10 </function>`<br><br>**Result:**<br>Index = 2 |
| **String GetAt** | This function reads a character from the specified position.<br>**Parameter:**<br>**str** - String<br>**index** - Zero-based index for the character being read<br><br>**Return value:**<br>The name of a string variable must be specified in which the character is to be stored.<br><br>**Syntax:**<br>`<function name="string.getat" return="<result string>"><string>, <index></function>`<br><br>**Example:**<br>`<let name="resStr" type="string" />`<br>`<function name="string.getat" return="resStr">_T"brown", 2</function>` |

| Function name | Description |
|---|---|
| **String SetAt** | This function writes a character to the specified position. The index must be smaller than the maximum text length.<br><br>**Parameter:**<br>**str** - String<br>**char** - Character which is to be written to the specified position<br>**index** - Zero-based index for the character string of the target variable<br><br>**Syntax:**<br>`<function name="string.setat" ><destination string>, <character string>, <index></function>`<br><br>**Example:**<br>`<let name="str" type="string" >br_wn</string>`<br>`<function name="string.setat">str, ">_T"o", 2 </function>` |
| **String Split** | This function deconstructs a string into a number of substrings and copies them into the string array specified. Separation is implemented at the specified separator. The separator is not saved in the substring.<br>This function expands the array automatically if the number of separators located is greater than the defined array size.<br><br>**Parameter:**<br>**str** - String<br>**char** - Name on a variable which includes the separator<br>**number** - Name on a variable which includes the number of generated substrings after the function has been performed.<br><br>**Return value:**<br>The name of a string array must be specified which includes the substrings after the function has been performed.<br><br>**Syntax:**<br>`<function name=" string.split" return="<result string array>"><string>, <char>, <number></function>`<br><br>**Example:**<br>`<let name="strlist" type="string" dim="2"/>`<br>`<let name="str_num" />`<br>`<function name="string.split" return="strlist">`<br>`_T"brown;green;blue;red", _T";", str_num </function>` |

# 1.11 Multitouch operation

## 1.11.1 Multitouch function

### Overview

When multitouch displays are introduced, it is necessary to adjust operation to the extended functionality of the display. The rigid positioning of softkeys, for example, is eliminated and replaced or supplemented by free positioning of buttons. Due to the diversity of design options for buttons, it no longer makes sense to use just one control for programming whose appearance is based exclusively on the design specifications of the operating software. Toggle controls, which only know two states, can be replaced by switches, for example, that show the respective state by means of an icon and react to tap or flick gestures.

Displays graphics can be given the capability of reacting to pan gestures. The **imagebox** control has been extended for this purpose.

**Note**

Graphics displayed in the paint tag still do not react to gestures.

Outside of the controls provided by the parser, finger gestures can be processed in the script, thus providing the option, for example, of offering new navigation strategies in the dialogs. Finger gestures could be used for zooming in to/out of dialog contents.

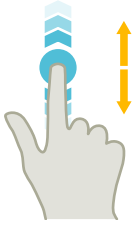
The Easy XML parser supports the following finger gestures:

### Finger gestures



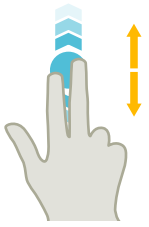

**Tap**

- Select window
- Select object (e.g. NC set)
- Activate entry field
  - Enter or overwrite value
  - Tap again to change the value



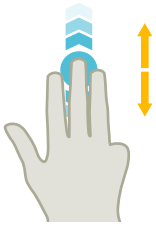
**Flick vertically with one finger**

- Scroll in lists (e.g. programs, tools, zero points)
- Scroll in files (e.g. NC program)



**Flick vertically with two fingers**

- Page-scroll in lists (e.g. ZO)
- Page-scroll in files (e.g. NC programs)



**Flick vertically with three fingers**

- Scroll to the start or end of lists
- Scroll to the start or end of files



**Flick horizontally with one finger**

- Scroll in lists with many columns



**Spread**

- Zoom in to graphic contents (e.g. simulation, mold making view)



**Pinch**

- Zoom out from graphic contents (e.g. simulation, mold making view)



**Pan with one finger**

- Move graphic contents (e.g. simulation, mold making view)
- Move list contents

The **gesture_event** tag and the **$gestureinfo** variable are used for handling finger gestures. The enable program actions for decoded finger gestures.

## 1.11.2 Programming finger gestures

### gesture_event tag

**Note**

This tag is only executed if the operating software's gesture control has been activated.

If the operating software recognizes a finger gesture, the parser provides the gesture information in the **$gestureinfo** variable and executes the **gesture_event** tag. The variable possesses the **GestureInfoStruct** data type and contains the following attributes:

| Attribute | Value | Meaning |
|---|---|---|
| **type** | 3 | Pan gesture |
| | 4 | Zoom out gesture |
| | 5 | Tap gesture |
| **flag** | 1 | Scaling factor changed |
| | 2 | Angle of rotation changed |
| **state** | 1 | Started |
| | 2 | Updated |
| | 3 | Completed |
| **item_data** | | Identification of the active control |
| | -1 | The gesture is not assigned to a control |
| | != -1 | The gesture was executed in a control to which this value was assigned during creation |
| **point** | | Position of the gesture |
| | point.x | X position of the gesture |
| | point.y | Y position of the gesture |
| **delta** | | Difference between the start of the gesture and the current event |
| | <Scaling difference> | When scaling factor is changed |
| | <Angular difference> | When angle of rotation is changed |

These attributes are predefined in the **struct_def.xml** file.

## 1.11.3 Gesture control for graphics

### Imagebox control variable

The following three extensions are available for the gesture control of graphics when using the **Imagebox** control variable:

| Attribute | Meaning/behavior |
|---|---|
| rotationangle | This attribute value indicates the angle at which the graphic is to be displayed. |
| setrotationmode | The **true** attribute value allows processing of the pinch gesture |
| setzoommode | The **true** attribute value allows you to zoom in to/out from or move a graphic in the display area. This default setting of this attribute is **false**. |

### Syntax

```
<property rotationangle="angle" />
<property setrotationmode="true/false" />
<property setzoommode="true/false" />
```

### Example of bitmap rotation

Rotate bitmap by 30.5 degrees clockwise:

```
<let name="image_box_pict_name" type="string" >pic1.bmp</let>
…
…
…
 <control name="image_view" xpos="100" ypos="123" width="300"
height="200" fieldtype="imagebox" refvar="image_box_pict_name"
hotlink="true" >
 <property item_data="1000" />
 <property rotationangle="30.5" />
 <property setrotationmode ="true" />
 </control>
```



## Example of bitmap zooming

Allowing zooming for a bitmap:

```
<control name="image_view" xpos="100" ypos="123" width="300"
height="200" fieldtype="imagebox" refvar="image_box_pict_name"
hotlink="true" >
 <property item_data="1000" />
 <property setzoommode="true" />
 </control>
```



## imageboxset control function

The properties can also be set using the **control.imageboxset** function.

The following commands are available:

| Command | Meaning/behavior |
|---|---|
| SetRotationAngle | The graphic is rotated around the angle indicated in **lparam1**. |
| SetRotationMode | The value of **lparam1** defines the evaluation of the pinch gesture with respect to the rotation.<br>Value equal to 1 - the gesture is processed by the control |
| SetZoomMode | If the value of **lparam1** is equal to 1, the pinch gesture for zooming in to/out from or moving the graphic is evaluated. |

## 1.11.4 Gesture processing

### Tag message

If the scaling factor is greater than 1.0, the image in the display area will be moved. If the factor is 1.0, this finger gesture can be used for browsing through a list of images, for example. In this case, the parser sends a message to the form, which can be evaluated in the **message** tag.

The **$message_par1** message parameter contains the Item_Data value of the control. The **$message_par2** message parameter signals the gesture's direction of movement.

The **$message_par2** can accept the following values:

- -1 roll left
- 1 roll right
- -2 roll up
- 2 roll down

### Example

```
…
…
<let name="image_box_pict_name" type="string" ></let>
<let name="pictindex" />
<let name="image_box_list" type="string" dim="4">
 "pic1.bmp",
 "pic2.bmp",
 "pic3.bmp",
 "pic4.bmp",
</let>

…
…
<control name="image_view" xpos="100" ypos="123" width="300"
height="200" fieldtype="imagebox" refvar="image_box_pict_name"
hotlink="true" >
 <property item_data="1000" />
 <property setzoommode="false" />
</control>
…
…
…
<!--
 -1 roll left
1 roll right
-2 roll up
2 roll down
 -->
<message>
 <if condition="$message_par1 == 1000">
```

```
  <then>

   <switch>
    <condition>$message_par2</condition>
    <case value="1">
     <op>
      pictindex = pictindex+1;
     </op>
     <if condition="pictindex > 3">
      <then>
       <op>
        pictindex = 0;
       </op>
      </then>
     </if>
    </case>
    <case value="-1">
     <op>
      pictindex = pictindex-1;
     </op>
     <if condition="pictindex < 0">
      <then>
       <op>
        pictindex = 3;
       </op>
      </then>
     </if>
    </case>
   </switch>
    <op>
     image_box_pict_name = image_box_list[$pictindex];
    </op>
   </then>
  </if>
 </message>
…
…
```

# 1.12 Configuring your own buttons

Buttons can be integrated into a form as action buttons or selection buttons.

## 1.12.1 Pushbutton

### Tag property

The pushbutton is a control element that can be used as a button or checkable button. The button can be designed by means of attribute definitions in the "softkey look & feel" style as well as in a customized style. The relevant attributes must be defined with the **property** tag.

The foreground or background color can be changed using the **color_fg**, **color_bk**, **color_fg_pressed** and **color_bk_pressed** attributes.

The **pressed/checked** or **released** states are represented by the values one or zero. A handler function must be indicated to evaluate a button's change of state. The handler function is indicated with the **function** attribute in the control tag.

The state of a checkable button can be determined by reading out the control variable or an assigned reference variable.

With a touch operation, the finger gestures are "pressed" and "released" only when the finger gesture is "let go".

### Syntax

```
<control name="name" xpos="x position" ypos="y position"
fieldtype="pushbutton" >
 <caption></caption>
</control>
```

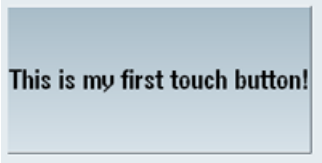
Or with handler function

```
<control name="name" xpos="x position" ypos="y position"
fieldtype="pushbutton" function="button_handler" hotlink="true" >
 <caption></caption>
</control>
…
…
…
 <function_body name="button_handler">
…
…
…
 </function_body>
```

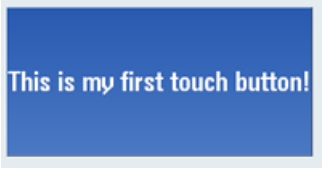
or

```
<control name="name" xpos="x position" ypos="y position"
fieldtype="pushbutton" refvar="button_state" hotlink="true" >
 <caption></caption>

 <property checkable="true" />
</control>
```




Not checked




Checked

## Example

```
<control name="name" xpos="x position" ypos="y position"
fieldtype="pushbutton" refvar="button_state" hotlink="true"
color_fg="#ff00ff" color_bk="#000eee">
 <property checkable="true" />
  <caption></caption>
 <property picture="sk_circ_grind_cg.png" alignment="left"
TextAlignedToPicture="true" scaled="true"/>
</control>
```



## 1.12.2 Functions of the pushbutton

### 1.12.2.1 Sub-tags for the pushbutton

#### Tag caption

The programmer uses the **caption** tag to specify the button text to be displayed for the "not pressed" state. In the default setting, the text is centered. The text alignment can be changed with the **alignment** attribute. The following attribute values can be specified:

| Attribute value | Alignment |
|---|---|
| left | Left-aligned |
| right | Right-aligned |
| top | Top |
| bottom | Bottom |
| center | Centered |

If a different text is to be displayed for the "pressed" state, a second caption instruction must be programmed with the **pressed** attribute and the **true** value.

#### Syntax

**Centered display**

```
<control name="name" xpos="x position" ypos="y position"
fieldtype="pushbutton" >
 <caption>Button text</caption>
</control>
```




**Left-aligned display**

```
<control name="name" xpos="x position" ypos="y position"
fieldtype="pushbutton" >
 <caption alignment= "left">Button text</caption>
</control>
```



**Pressed display**

```
<control name="name" xpos="x position" ypos="y position"
fieldtype="pushbutton" >
 <caption pressed="true">Button text pressed</caption>

</control>
```

### 1.12.2.2 Properties for the pushbutton

Additional properties are assigned to the control with the **property** tag.

#### Determining button properties

The **checkable** attribute enables the button to be used as checkable pushbutton. To do so, the value of the attribute must be set to **true**.

**Syntax**

```
<control name="name" xpos="x position" ypos="y position"
fieldtype="pushbutton" >
 <property checkable="true/false" />
</control>
```

#### Disabling a switch

The **disabled** attribute controls whether or not the button can be operated. If the value is **true**, control actions are not processed.

Changes in state caused by an assigned reference variable result in updating of the button.

**Syntax**

```
<property disabled="true/false" />
```

**Example**

```
<control name="name" xpos="x position" ypos="y position"
fieldtype="pushbutton" >
 <caption alignment="left">Button text</caption>
 <property disabled="true " />
</control>
```

#### Assigning icons

The predefined design can be covered over by specifying your own icons or button colors. An icon can be assigned to the button for each of the states "not pressed", "pressed" and "disabled". If there is no assignment for the states "pressed" or "disabled", the control displays the icon for the "not pressed" state.

Other attributes control the alignment, scaling and transparency of each icon.

| Attribute | Meaning/behavior |
|---|---|
| Picture | Icon in foreground<br>Name of the icon for the "not pressed" state |
| PicturePressed | Icon in foreground<br>Name of the icon for the "pressed" state |
| PictureDisabled | Icon in foreground<br>Name of the icon for the "disabled" state |

| Attribute | Meaning/behavior |
|---|---|
| BackgroundPicture | Icon in background<br>Name of the icon for the "not pressed" state |
| BackgroundPicturePressed | Icon in background<br>Name of the icon for the "pressed" state |
| BackgroundPictureDisabled | Icon in background<br>Name of the icon for the "disabled" state |

## Alignment attribute

Further attributes can be specified in the tag whose values refer to the **alignment** of the listed icon assignments:

| Attribute value | Alignment |
|---|---|
| left | Left-aligned |
| right | Right-aligned |
| top | Top |
| bottom | Bottom |
| center | Centered |
| stretch | Background icon:<br>If the **stretch** value is used, the parser scales the icon to the rectangular area of the button.<br>Any desired outline can be generated for the button by declaring the transparent color with the **transparent** attribute. |

### 1.12.2.3 Control variables for the pushbutton

Button properties can be subsequently changed by assigning new values to the attribute variables listed below. The values are assigned in an operation instruction by specifying the control name followed by the control variable name. The two names must separated by a period.

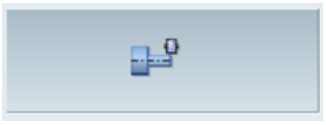
## Syntax

<Control-Name>.<Control-Variable>

| Control variable | Meaning/behavior |
|---|---|
| backgroundpicture | Variable type: String<br>Name of background icon |
| backgroundpicturedisabled | Variable type: String<br>Name of background icon for the disabled state |
| backgroundpicturepressed | Variable type: String<br>Name of background icon for the pressed state |
| picture | Variable type: String<br>Name of foreground icon |
| picturedisabled | Variable type: String<br>Name of foreground icon for the disabled state |
| picturepressed | Variable type: String<br>Name of foreground icon for the pressed state |
| picture_textalignedtopicture | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>Button text is aligned on the icon |
| picturedisabled_textalignedtopicture | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>Button text is aligned on the "disabled state" icon |
| picturepressed_textalignedtopicture | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>Button text is aligned on the "pressed state" icon |
| backgroundpicture_keepaspectratio | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The aspect ratio of the background icon is retained or ignored |

| Control variable | Meaning/behavior |
|---|---|
| backgroundpicturedisabled_keepaspectratio | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The aspect ratio of the "disabled state" background icon is retained or ignored |
| backgroundpicturepressed_keepaspectratio | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The aspect ratio of the "pressed state" background icon is retained or ignored |
| picture_keepaspectratio | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The aspect ratio of the icon is retained or ignored |
| picturedisabled_keepaspectratio | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The aspect ratio of the "disabled state" icon is retained or ignored |
| picturepressed_keepaspectratio | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The aspect ratio of the "pressed state" icon is retained or ignored |
| backgroundpicture_scaled | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The background icon is adjusted to the dimensions of the button |
| backgroundpicturedisabled_scaled | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The "disabled state" background icon is adjusted to the dimensions of the button |

| Control variable | Meaning/behavior |
|---|---|
| backgroundpicturepressed_scaled | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The "pressed state" background icon is adjusted to the dimensions of the button |
| picture_scaled | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The icon is adjusted to the dimensions of the button |
| picturedisabled_scaled | Variable type: Bool<br>Value:<br>1 = true<br>0 = false<br>The "disabled state" icon is adjusted to the dimensions of the button |
| picturepressed_scaled | Variable type: Bool<br>The "pressed state" icon is adjusted to the dimensions of the button |
| backpicture_alignment | Variable type: String<br>See attribute alignment |
| backgroundpicturedisabled_alignment | |
| backgroundpicturepressed_alignment | |
| picture_alignment | |
| picturedisabled_alignment | |
| picturepressed_alignment | |
| caption | Variable type: String<br>"Not pressed state" button text |
| captionpressed | Variable type: String<br>"Pressed state" button text |
| caption_alignment | Variable type: String<br>See attribute alignment |
| captionpressed_alignment | |
| captiondisabled_alignment | |
| disabled | Variable type: Bool<br>Value:<br>1 = true - pushbutton disabled<br>0 = false - pushbutton can be operated |



Picture



PicturePressed



PictureDisabled



BackgroundPicture + Picture

### TextAlignedToPicture attribute

If the value of the attribute is **true**, the text is aligned relative to the icons.

```
<property picture="sk_circ_grind_cg.png" alignment="left"
TextAlignedToPicture="true" />
```




### Scaled attribute

If the value of the attribute is **true**, the dimensions of the image are adjusted to the dimensions of the button in such a way that 50% of the height or width of the button at most is made available for the graphic.

```
<property picture="sk_circ_grind_cg.png" alignment="left"
TextAlignedToPicture="true" scaled="true"/>
```




### Stretch attribute (only effective for background icons)

If the value of the attribute is **true**, the dimensions of the image are adjusted to the dimensions of the button.

```
<property backgroundpicture="f:\appl\pbutton.png"
alignment="stretch" />
```



```
<property backgroundpicture="f:\appl\pbutton.png"
alignment="stretch" transparent="#ffffff"/>
```



```
<caption alignment="left" >This is my first touch button!</caption>
 <property picture="sk_circ_grind_cg.png" alignment="left"
TextAlignedToPicture="true" scaled="true"/>
 <property backgroundpicture="pbutton.png" alignment="stretch"
transparent="#ffffff"/>
```




## 1.12.3 Switch on/off

A switch control is a graphic element that signals one of two states by means of an icon. This control can be operated by touch or by using a mouse.

The state adopted can be stored in a reference variable or the change of state can be determined in a function assigned to the control. The function is assigned with the **function** attribute in the control tag.

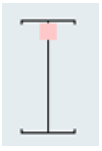
The **alignment** attribute enables the vertical or horizontal alignment of the button.

This control has no standard design. If no icons are assigned to the control, only the bounding box and switch position (shown by a dot) are displayed.

### Syntax

```
<control name="name" xpos = "x position" ypos="y position"
width="width" height="height" fieldtype="switch" alignment="hr/vr">
 <property left_position="value" />
 <property right_position="value" />
</control>
```

### Alignment attribute



Vertical switch: Value **vr**



Horizontal switch: Value **hr**

## 1.12.4 Functions of the switch

### 1.12.4.1 Properties for the switch

### Tag property

The following switch positions can be assigned to the control with the **property** tag:

| Attribute | Meaning/behavior |
|---|---|
| left_position | The value of the attribute is assigned to the control variable if the switch was moved to the left. |
| right_position | The value of the attribute is assigned to the control variable if the switch was moved to the right. |
| upper_position | The value of the attribute is assigned to the control variable if the switch was moved up. |
| lower_position | The value of the attribute is assigned to the control variable if the switch was moved down. |
| disabled | The attribute controls whether or not the switch can be operated.<br>If the value is **true**, a control action is not evaluated.<br>Changes in state caused by an assigned reference variable result in updating of the switching state. |

The final switch design must be determined by specifying further attributes. These attributes must be defined together with the attribute **left_position**, **right_position**, **upper_position** or **lower_position**.

Any desired outline can be generated for the switch by declaring one color as a transparent color with the **transparent** attribute.

| Attribute | Meaning/behavior |
|---|---|
| picture | Icon in foreground Name of the icon for the "not pressed" state |
| pictureDisabled | Icon in foreground Name of the icon for the "disabled" state |
| transparent | The attribute value includes the color value of the transparent color.<br>This color value is used for all icons assigned to the switch. |
| caption | The attribute value includes the text associated with the switch position. This text is displayed on the element and limited by the dimensions of the switch. |

### 1.12.4.2 Control variables for the switch

Switch properties can be subsequently changed by assigning new values to the control variables listed below. The values are assigned in an operation instruction by specifying the control name followed by the control variable name. The two names must separated by a period.

## Syntax

<Control-Name>.<Control-Variable>

| Control variable | Meaning/behavior |
|---|---|
| Left_position | Variable type: Int<br>The value of the attribute is assigned to the control variable if the switch was moved to the left. |
| Right_position | Variable type: Int<br>The value of the attribute is assigned to the control variable if the switch was moved to the right. |
| Lower_position | Variable type: Int<br>The value of the attribute is assigned to the control variable if the switch was moved down. |
| Upper_position | Variable type: Int<br>The value of the attribute is assigned to the control variable if the switch was moved up. |
| Picture_left_position | Variable type: String<br>Name of icon for the left-hand position |
| Picture_right_position | Variable type: String<br>Name of icon for the right-hand position |
| Picture_upper_position | Variable type: String<br>Name of icon for the upper position |
| Picture_lower_position | Variable type: String<br>Name of icon for the lower position |
| Picturedisabled_left_position | Variable type: String<br>Name of icon for the left-hand position "disabled state" |
| Picturedisabled_right_position | Variable type: String name of icon for the right-hand position "disabled state" |
| Picturedisabled_upper_position | Variable type: String<br>Name of icon for the upper position "disabled state" |
| Picturedisabled_lower_position | Variable type: String<br>Name of icon for the lower position "disabled state" |
| Caption_left_position | Variable type: String<br>Text in the left-hand position |
| Caption_right_position | Variable type: String<br>Text in the right-hand position |

| Control variable | Meaning/behavior |
|---|---|
| Caption_upper_position | Variable type: String<br>Text in the upper position |
| Caption_lower_position | Variable type: String<br>Text in the lower position |
| disabled | Variable type: Bool<br>Value:<br>1 = true - switch disabled<br>0 = false - switch can be operated |

**Horizontal display**

```
<control name="name" xpos="x position" ypos="y position"
width="width" height="height" fieldtype="switch" alignment="hr">
 <property left_position="value" picture="name" />
 <property right_position="value" picture="name" />
</control>
```

**Vertical display**

```
<control name="name" xpos="x position" ypos="y position"
width="width" height="height" fieldtype="switch" alignment ="vr">
 <property upper_position="value" picture="name" />
 <property lower_position="value" picture="name" />
</control>
```

**Or assign handler function**

```
<control name="name" xpos="x position" ypos="y position"
width="width" height="height" fieldtype="switch"
function="switch_handler" hotlink="true" alignment ="vr">
</control>
```

## Example



icon_left.bmp



icon_right.bmp

```
<let name="switch_state" />
<control name="main_switch" xpos="100" ypos="53" width="40"
height="30" fieldtype="switch" refvar="switch_state" hotlink="true"
alignment="hr">
 <property left_position="10" picture="icon_left.bmp" />
 <property right_position="11" picture="icon_right.bmp" />
```

```
</control>
```

Assigning properties by means of control variables

```
<control name="main_switch" xpos = "450" ypos="340" width="20"
height="46" fieldtype="switch" refvar="switch_statebf"
hotlink="true" alignment="vr">
 <property upper_position="1" />
 <property lower_position="0" />
</control>
…
…
…
<op>
  main_switch.transparent = #ffffff;
  main_switch.picture_upper_position = _T"switch_up.png";
  main_switch.picture_lower_position = _T"switch_bottom.png";
  main_switch.disable= 1;
</op>
```

## 1.12.5 Radio button

Radio buttons enable the selection of one of a number of options. A button must be created for each option. All radio buttons belonging to one form, groupbox or scroll area interact with one another in such a way that only one button is marked as selected. The button states are transferred to the control variables.

### Syntax

```
<control name="name" xpos="x position" ypos="y position"
width="width" height="height" fieldtype="radiobutton"  >
 <caption>button text</caption>
</control>
```

## 1.12.6 Checkbox

Checkboxes enable the selection of a number of options. A button must be created for each option. The state of the checkbox is transferred to the control variable.

### Syntax

```
<control name="name" xpos="x position" ypos="y position"
width="width" height="height" fieldtype="checkbox"  >
 <caption>button text</caption>
</control>
```

## 1.12.7 Groupbox

A groupbox optically encloses a group of controls with a frame and and a title. It groups logically related controls that are only intended to interact within the group. The coordinates of the embedded controls refer to the upper left-hand corner below the title line.

All of the controls belonging to the group are embedded into the control tag as sub-controls.

When assigning the embedded control names and the **Item_Data** value, it is important to note that they must be unique with respect to all controls belonging to the form.

The title of the groupbox is specified with the **caption** tag.

### Syntax

```
<control name="name" xpos="x position" ypos="y position"
width="width" height="height" fieldtype="groupbox">
 <caption>caption text</caption>
  <control>…</control>
…
…
…
</control>
```

## 1.12.8 Scroll area

A scroll area is used to display controls within a specified area. The coordinates of the embedded controls refer to the upper left-hand corner of the scroll area. If controls are created outside of the visible area, moving the visible area is enabled. All of the controls belonging to the area must be embedded into the control tag as sub-controls.

When assigning the embedded control names and the **Item_Data** value, it is important to note that they must be unique with respect to all controls belonging to the form.

### Syntax

```
<control name="name" xpos="x position" ypos="y position"
width="width" height="height" fieldtype="scrollarea">
  <control>…</control>
…
…
…
</control>
```

### Touch operation

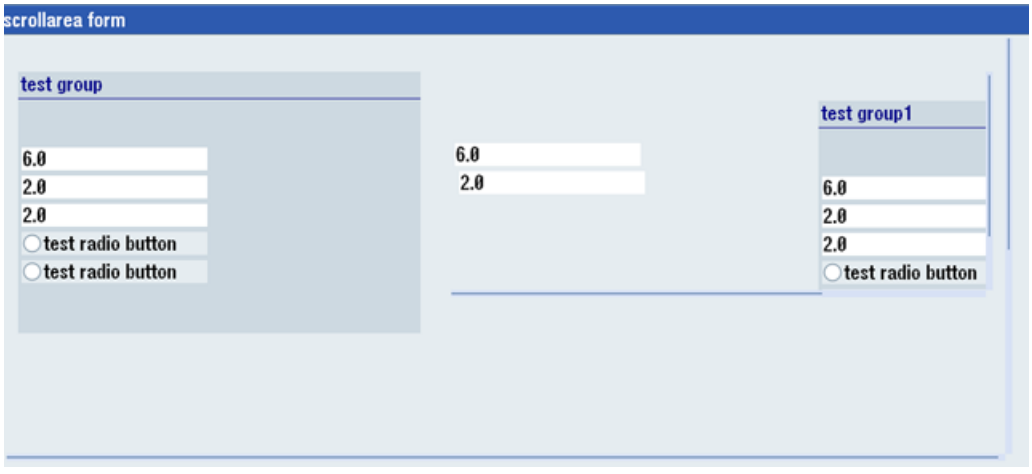
If focus is placed on a control that is currently not completely visible, the visible area is moved until the control can be displayed in full.

### Tab gesture

This gesture is forwarded to the script if the gesture cannot be assigned to an embedded control.

### Example

**Nested scroll areas**




```
<let name="CB1" >1</let>
```

```
<let name="RB1" />

<form name="scrollarea_form">
 <init>
  <caption>scrollarea form</caption>
   <data_access type="true" />
    <control name= "c_scroll" xpos = "2" ypos="24" width="520"
height="300" fieldtype="scrollarea" >
    <control name= "c_group" xpos = "6" ypos="24" width="208"
height="186" fieldtype="groupbox" >
    <caption>test group</caption>
    <control name = "c18" xpos = "2" ypos = "54" refvar="nck/Channel/
Parameter/R[1]" hotlink="true" />
    <control name = "c19" xpos = "2" ypos = "74" refvar="nck/Channel/
Parameter/R[2]" hotlink="true" />
    <control name = "c20" xpos = "2" ypos = "94" refvar="nck/Channel/
Parameter/R[3]" hotlink="true" />
    <control name="radiobg" xpos = "2" ypos="114"
fieldtype="radiobutton" >
    <caption>test radio button</caption>
    <property backgroundpicture="f:\appl\cycle_my1.png" />
   </control>
   <control name="radiobg1" xpos = "2" ypos="134"
fieldtype="radiobutton" >
    <caption>test radio button</caption>
    <property backgroundpicture="f:\appl\cycle_my1.png" />
   </control>
  </control>
 <control name= "c_scroll_emb" xpos = "230" ypos="24" width="280"
height="160" fieldtype="scrollarea" >

 <control name = "c18emb" xpos = "2" ypos = "54" refvar="nck/Channel/
Parameter/R[1]" hotlink="true" />
 <control name = "c19emb" xpos = "4" ypos = "74" refvar="nck/Channel/
Parameter/R[2]" hotlink="true" />
 <control name = "c20emb" xpos = "3" ypos = "194" refvar="nck/Channel/
Parameter/R[3]" hotlink="true" />
 <control name= "c_group1" xpos = "190" ypos="24" width="208"
height="186" fieldtype="groupbox" >
 <caption>test group1</caption>
 <control name = "c18gemb" xpos = "2" ypos = "54" refvar="nck/Channel/
Parameter/R[1]" hotlink="true" />
 <control name = "c19gemb" xpos = "2" ypos = "74" refvar="nck/Channel/
Parameter/R[2]" hotlink="true" />
 <control name = "c20gemb" xpos = "2" ypos = "94" refvar="nck/Channel/
Parameter/R[3]" hotlink="true" />
 <control name="radiobggemb" xpos = "2" ypos="114"
fieldtype="radiobutton" >
  <caption>test radio button</caption>
  <property backgroundpicture="f:\appl\cycle_my1.png" />
</control>
```

```
<control name="radiobg1emb" xpos = "2" ypos="134"
fieldtype="radiobutton" >
 <caption>test radio button</caption>
  <property backgroundpicture="f:\appl\cycle_my1.png" />
 </control>
 </control>
 </control>
<control name = "c22" xpos = "2" ypos = "400" refvar="nck/Channel/
Parameter/R[12]" hotlink="true" />
 <control name="ChB1" xpos="208" ypos="430" width="50" height="30"
fieldtype="checkbox" refvar="PLC/M100.7" hotlink = "true"
color_bk="#00ffff">
  <caption>Check1</caption>
 </control>
  <control name="ChB2" xpos="208" ypos="460" width="58" height="30"
fieldtype="checkbox" refvar="PLC/M100.6" hotlink = "true"
color_bk="#00ffff">
  <caption>Check2</caption>
 </control>
  <control name="Radio1" xpos="208" ypos="490" width="40"
height="30" fieldtype="radiobutton" refvar="PLC/M100.0" hotlink =
"true" color_bk="#00ffff">
   <caption>Radio1</caption>
  </control>
   <control name="Radio2" xpos="208" ypos="520" width="45"
height="30" fieldtype="radiobutton" refvar="PLC/M100.1" hotlink =
"true" color_bk="#00ffff">
    <caption>Radio2</caption>
  </control>
   <control name="Radio3" xpos="208" ypos="550" width="50"
height="30" fieldtype="radiobutton" refvar="PLC/M100.2" hotlink =
"true" >
    <caption>Radio3</caption>
   </control>
    <control name="VChB1" xpos="8" ypos="430" width="50" height="30"
refvar="PLC/MB100" hotlink = "true" color_bk="#00ffff"/>
    <control name="VChB2" xpos="8" ypos="465" width="53" height="30"
refvar="PLC/MB100" hotlink = "true" color_bk="#00ffff"/>
   </control>
   <control name="ChB3" xpos="208" ypos="340" width="60" height="30"
fieldtype="checkbox" refvar="CB1" >
    <caption>Check3</caption>
   </control>
   </init>
  <timer>
  </timer>
</form>
```

# 1.13 Sidescreen application

## 1.13.1 Easy XML in the Sidescreen

The Easy XML scripting language can also be used to create dialogs in the Sidescreen area. The Sidescreen components **Page** and **Widget** are available for displaying the dialogs. The connection is made via the **slsidescreen.ini** file. Sidescreen components are automatically started when the operating software starts up and are only terminated when the system shuts down. This means that the parser loads the assigned scripts once when the Sidescreen component is activated for the first time.

The dimension of the Sidescreen area is determined by the layout parameters from the **slsidescreen_<screen resolution>.ini** file. To enable independent configuration, the usable width for Easy XML scripts is 276 pixels. The usable height of an Easy XML dialog depends on the Sidescreen component used. For a page, 768 pixels are available. For a widget, the height is determined by the Sidescreen management and can be queried with the **GETHMIResolution** function.

The script parser expects a menu to activate a shape or branch to other shapes. The main menu must be named **main**. Sidescreen does not support softkey tags, so that you need to navigate to other shapes using an operator control for the shape.

The number of Sidescreen pages or Sidescreen widgets is not limited by the parser.

## 1.13.2 Integrating Sidescreen dialogs

The pages or widgets are integrated using the **slsidescreen.ini** file.

Pages are entered in the **[Sidescreen]** section. Each page must be specified with the keyword **PAGE** followed by the sequential page number and the required attributes. The **name** attribute defines the object name of the page. The **implementation** attribute specifies the implementation library and the class name. Both attributes must be separated by a comma.
**SlSideScreenPage** type pages can include Sidescreen elements. This is done via the "**ELEMENT**" entries. The rule for creating an **ELEMENTxxx** element line corresponds to the rule for creating a page. If a page should only contain dialogs that were configured with Easy XML, **slagmforms.SlEESideScreenPage** should be specified as value for the **implementation** attribute.

Widgets can be added to a Sidescreen element in the section **[Element_<name>]**.

The rule for creating a **WIDGETxxx** widget line corresponds to the rule for creating a page.

If an Easy XML Sidescreen widget is to be activated, **slagmforms.SlEESideScreenWidget** should be specified as the value for the **implementation** attribute.

### Syntax

```
ELEMENT002= name:=<name>, implementation:=SlSideScreenElement

…
…

[Element_<name>]
WIDGET001= name:=<Widget Name>, implementation:=
slagmforms.SlEESideScreenWidget
```

The widget identifier is used by default to form the main module name.

### Example

```
PAGE004= name:=sidescreen_proginfluence,
implementation:=slagmforms.SlEESideScreenPage
```

Main module name: sidescreen_proginfluence.xml

### Additional properties

Additional properties can be assigned to the pages or widgets by entering a section named **PAGE_<pagename>**, **ELEMENT_<elmentname>** or **WIDGET_<widgetname>** in the file **slsidescreen.ini**.

The following properties can be assigned to a page, an element or a widget:

| Attribute | Meaning/behavior |
|---|---|
| TextId | Language-independent identifiers of the text |
| TextFile | Text file name |
| TextContext | Text context EASY_XML |
| Icon | Name of the icon |

| Property | Meaning/behavior |
|---|---|
| maskPath | The property specifies the main module name to use. |
| maskName | The property specifies the main menu name to use. |
| focusable | The property determines whether the input focus can be set to the element. |

### Example

```
[Page_sidescreen_proginfluence]
PROPERTY001= name:=focusable, type:=bool, value:="true"
Icon=sidescreen_proginfluence.png

[PAGE_sidescreen_proginfluence]
Icon= sidescreen_proginfluence.png
PROPERTY001= name:=focusable, type:=bool, value:="true"
PROPERTY002= name:=maskPath, type:=QString, value:="
sidescreen_proginfluence.xml"
PROPERTY003= name:=maskName, type:=QString, value:="main"
```

## 1.13.3 Language and text management

To manage the language-independent texts, each component can be assigned a text file. The text file name is formed from the name of the component, the language version and the extension ".ts".

**EASY_XML** should be used as the text context.

### Syntax

<name>_<language>.ts

### Example

```
PAGE004= name:=sidescreen_proginfluence,
implementation:=slagmforms.SlEESideScreenPage
```

Text file name: sidescreen_proginfluence_**eng.ts**

If no text file is specified, the parser searches for the text identifier in the **oem_xml_screens_xxx.ts** default text file.

## 1.13.4 Sidescreen components

### 1.13.4.1 Sidescreen element

If a page is linked to the default implementation, elements can be assigned to this page. A **[Page_<page_name>]** section is to be created for this and all elements associated with the page are listed.

### Example

```
[Sidescreen]
PAGE001= name:=<page_name>, implementation:=SlSideScreenPage

[Page_<page_name>]

ELEMENT<number>= name:=<Element name>,
implementation:=SlSideScreenElement
```

Each element requires an **[Element_<element name>]** section in which the properties and associated widgets are listed.

```
[Element_<element_name>]
WIDGET001= name:=<widget_name>, implementation:= <implementation>
```

### 1.13.4.2 Sidescreen widget

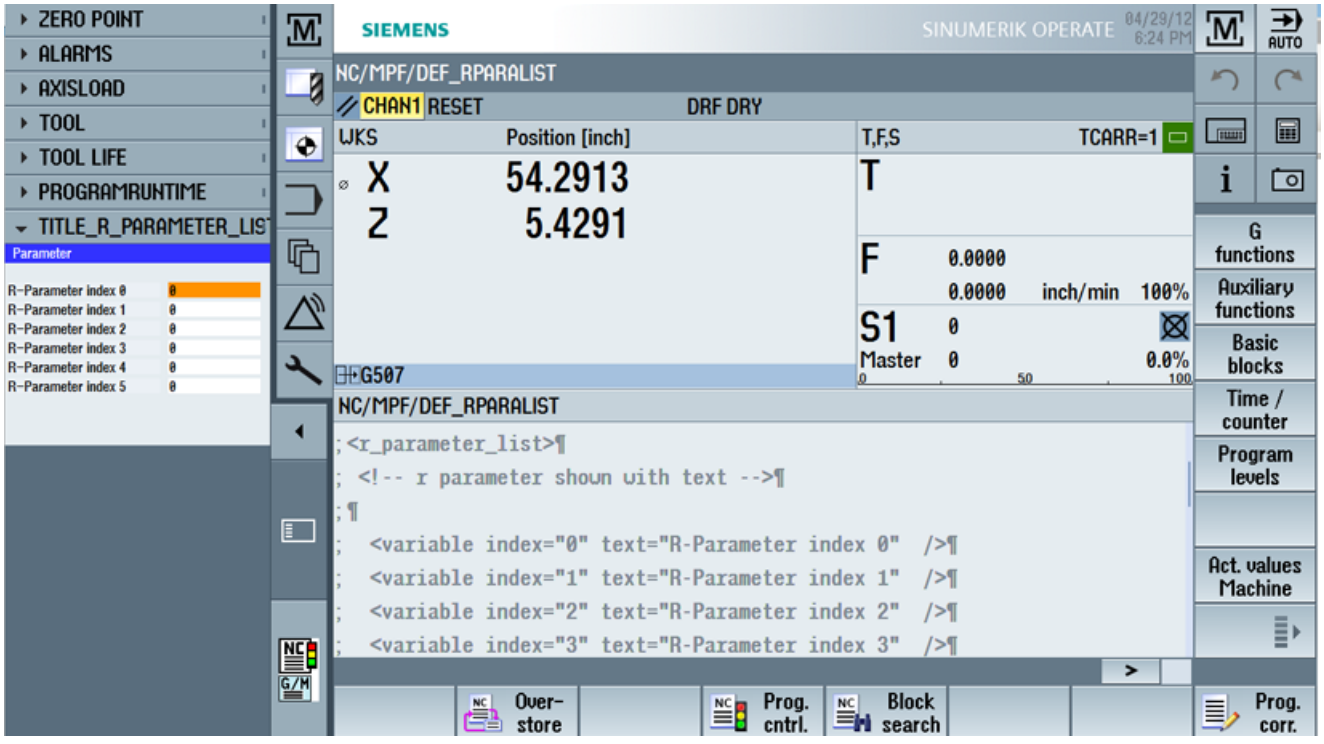
A widget is assigned an area by the Sidescreen management in which the configured shapes can be displayed. By default, a widget does not obtain an input focus so that fields cannot be edited. This behavior can be changed by specifying the **focusable** attribute. When the widget opens, the parser opens the form associated with the main menu. Within the form, the user can branch out to other menus with a navigation instruction.

## Example

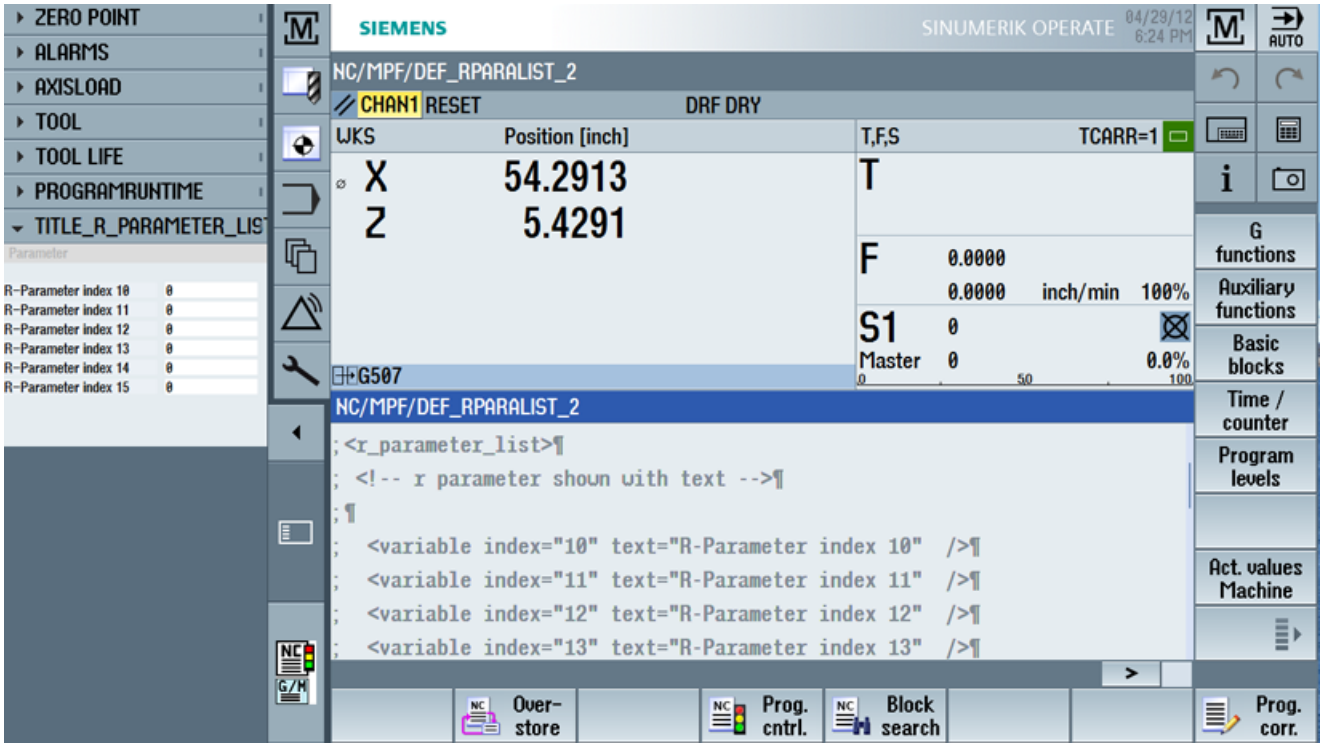
The Easy XML script searches for the description for the data content of the widget in the active part program. In this specific example, the description of a list with R parameters is expected that are to be displayed. Each parameter can be assigned a description text.

#### Toolbox example

Custom Screen Sample\
side_screen_examples\sidescreenwidget\displayRparameter\rparameter_widget.xml




If another part program is selected, the widget is automatically reworked.


ZERO POINT
ALARMS
AXISLOAD
TOOL
TOOL LIFE
PROGRAMRUNTIME
TITLE_R_PARAMETER_LIST
Parameter
R-Parameter index 10 0
R-Parameter index 11 0
R-Parameter index 12 0
R-Parameter index 13 0
R-Parameter index 14 0
R-Parameter index 15 0
SIEMENS
SINUMERIK OPERATE
04/29/12
6:24 PM
NC/MPF/DEF_RPARALIST_2
CHAN1 RESET
DRF DRY
WKS
Position [inch]
X 54.2913
Z 5.4291
T,F,S
TCARR=1
T
F 0.0000
0.0000 inch/min 100%
S1 0
Master 0 0.0%
G507
NC/MPF/DEF_RPARALIST_2
;<r_parameter_list>¶
; <!-- r parameter shoun uith text -->¶
;¶
; <variable index="10" text="R-Parameter index 10" />¶
; <variable index="11" text="R-Parameter index 11" />¶
; <variable index="12" text="R-Parameter index 12" />¶
; <variable index="13" text="R-Parameter index 13" />¶
AUTO
G functions
Auxiliary functions
Basic blocks
Time / counter
Program levels
Act. values Machine
Over-store
Prog. cntrl.
Block search
Prog. corr.

### 1.13.4.3 Sidescreen page

A page, which is started via the **slagmforms.SlEESideScreenPage** implementation, makes available the entire Sidescreen area of a form. It has no title line so that a heading cannot be configured with the Caption statement. By default, a page does not obtain an input focus so that fields cannot be edited. This behavior can be changed by specifying the **focusable** attribute. When the page opens, the parser opens the form associated with the main menu. Within the form, the user can branch out to other menus with a navigation instruction.

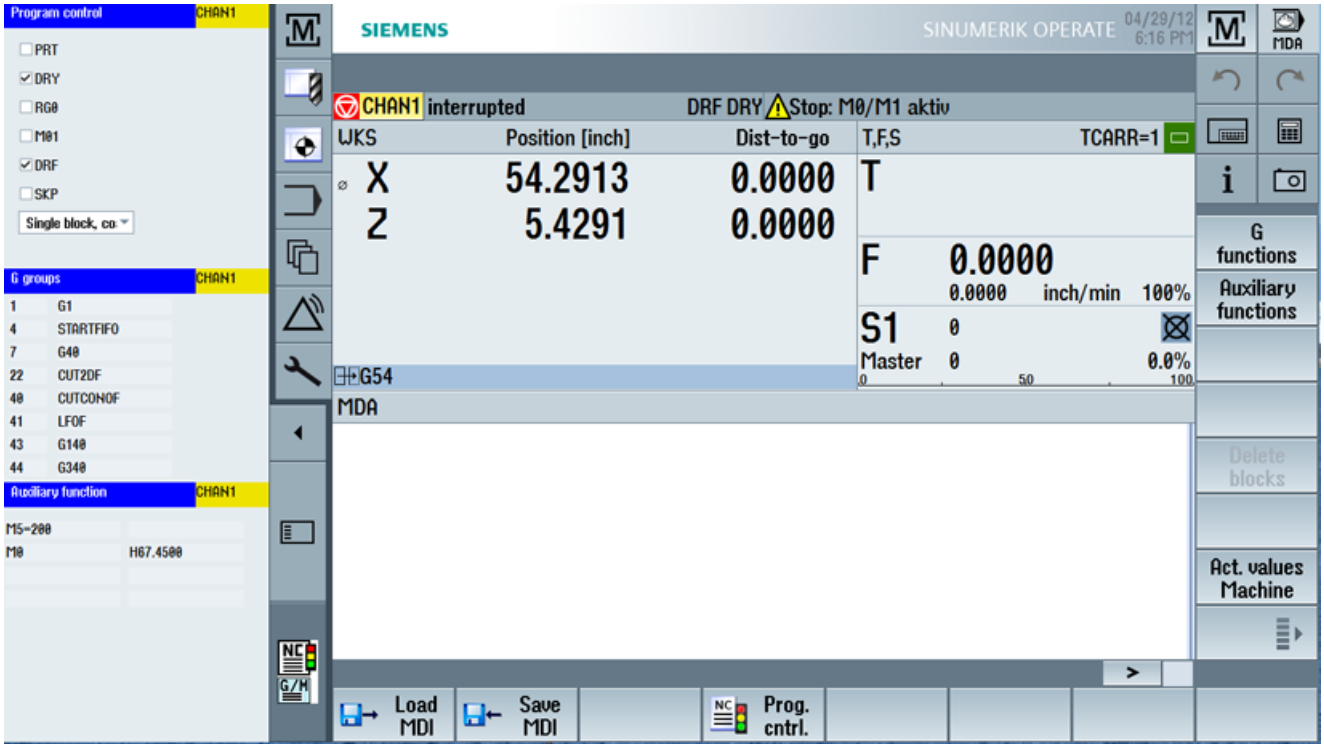
## Example

The Sidescreen page is displayed as page which only shows an Easy XML form.

In this example, additional information is displayed in the Sidescreen area.

#### Toolbox example

Custom Screen Sample\side_screen_examples\sidecreenpage\sidescreen_proginfluence.xml




```
[Sidescreen]
PROPERTY001= name:=minimizable, type:=bool, value:="true"
PROPERTY002= name:=buttonBarVisible, type:=bool, value:="true"
PAGE001= name:=WidgetsPage, implementation:=SlSideScreenPage
PAGE004= name:=sidescreen_proginfluence,
implementation:=slagmforms.SlEESideScreenPage
```

```
[Page_sidescreen_proginfluence]
PROPERTY001= name:=focusable, type:=bool, value:="true"
Icon=sidescreen_proginfluence.png
```

Text file: sidescreen_proginfluence_deu.ts

# 1.14 Dialog selection via PLC hardkeys

## Field of application

The following functions can be initiated by the PLC in the operating software:

- Select an operating area
- Select certain scenarios within operating areas
- Execute functions configured at softkeys

## Hardkeys

All keys - also the PLC keys - are subsequently referred to as hardkeys. A maximum of 254 hardkeys can be defined. The following allocation applies:

| Key number | Application |
|---|---|
| Key 1 – key 9 | Keys on the operator panel front |
| Key 10 – key 49 | Reserved |
| Key 50 – key 254<br>Key 50 – key 81 | PLC keys:<br>Reserved for OEMs |
| Key 255 | Pre-assigned with control information |

Hardkeys 1 - 9 are pre-assigned as follows:

| Key designation | | Action / effect |
|---|---|---|
| HK1 | MACHINE | Selects "Machine" operating area, last dialog |
| HK2 | PROGRAM | Selects "Program" operating area, last dialog or last program |
| HK3 | OFFSET | Selects "Parameter" operating area, last dialog |
| HK4 | PROGRAM MANAG-ER | Selects "Program" operating area, "Program manager" root screen<br>No function |
| HK5 | ALARM | Selects "Diagnostics" operating area, "Alarm list" dialog |
| HK6 | CUSTOM | Selects "Custom" operating area |

## Configuration

The configuring is realized in the systemconfiguration.ini configuration file in the section [keyconfiguration]. Each line defines what is known as a hardkey event. A hardkey event is the n-th actuation of a specific hardkey. For example, the second and third actuation of a specific hardkey can result in different responses.

The entries in the systemconfiguration.ini configuration file can be overwritten with user-specific settings. The directories [System user directory]/cfg and [System oem directory]/cfg are available for this purpose.

The lines for configuring the hardkey events have the following structure:

```
KEYx.n = area:=area, dialog:=dialog, screen:=screen, forms:=form,
menus:=menu, action:=menu.action, cmdline:=cmdline
KEYx.n = area:=area, dialog:=dialog, cmdline:=cmdline,
action:= action
```

x: Number of the hardkey, range of values: 1 – 254

n: Event number – corresponds to the nth actuation of the hardkey, range of values: 0 – 9

## Requirement

The PLC user program must fulfill the following requirement:

Only one hardkey is processed. As a consequence, a new request can only be set if the operating software has acknowledged the previous request. If the PLC user program derives the hardkey from an MCP key, it must provide sufficient buffer storage of the key(s) to ensure that no fast keystrokes are lost.

## PLC interface

An area to select a hardkey is provided in the PLC interface. The area is in DB19.DBB10. Here, the PLC can directly specify a key value of between 50 and 254.

Acknowledgment by the operating software takes place in two steps. This procedure is necessary so that the operating software can correctly identify two separate events if the same key code is entered twice consecutively. In the first step, control information 255 is written to byte DB19.DBB10. This defined virtual key stroke enables the HMI to identify every PLC key sequence uniquely. The control information is of no significance to the PLC user program and must not be changed. In the second step, the actual acknowledgment takes place with respect to the PLC by clearing DB19.DBB10. From this point in time, the PLC user program can specify a new hardkey. In parallel, the actual hardkey request is processed in the operating software.

## Example

Configuration file:

```
; configuration of OP hardkeys (KEY1-KEY9) and
; PLC hardkeys (KEY50-KEY254)
[keyconfiguration]
; MACHINE key (hardkey block)
KEY1.0 = area:=AreaMachine, dialog:=SlMachine
; PROGRAM key (hardkey block)
KEY2.0 = area:=AreaProgramEdit
; OFFSET key (hardkey block)
KEY3.0 = area:=AreaParameter
; PROGRAM MANAGER key (hardkey block)
KEY4.0 = area:=AreaProgramManager
; ALARM key (hardkey block)
```

```
KEY5.0 = area:=AreaDiagnosis, dialog:=SlDgDialog,
cmdline:="-slGfwHmiScreen SlDgAeAlarmsScreen"
; CUSTOM (hardkey block)
KEY6.0 = area:=Custom
; MACHINE key (optional, Shift + F10)
KEY7.0 = area:=AreaMachine, dialog:=SlMachine,
cmdline:="-MKey"
; MENU USER (hardkey block, Ctrl + F10)
KEY10.0 = area:=MenuUser
; MENU FUNCTION (hardkey block, Ctrl + Shift + F10)
KEY11.0 = area:=MenuFunction
KEY50.0 = area:=CustomXML, dialog:=SlEECustomDialog
```

The area and dialog identifiers can be found in the systemconfiguration.ini from [System Siemens directory]/cfg.

If only the **SlEECustomDialog** is used, the parser expects the **xmldial.xml** file in the application directory as well as a menu tag with the name **main**. Other file names or main menu names can be declared by adding the **cmdline** key. The **-mainModule** parameter specifies the XML description to be loaded. The parser expects the main menu in this script. The **-entry** parameter specifies the name of the main menu. If this parameter is missing, the parser uses the name **main** to start the function. The **-conf** parameter with the value **slagmdialog.hmi** must always be included.

Here is an example:

```
KEY50.0 = area:=CustomXML, dialog:=SlEECustomDialog,
cmdline:="-conf slagmdialog.hmi -mainModule restore.xml -entry cmc2"
KEY51.0 = area:=CustomXML, dialog:=SlEECustomDialog,
cmdline:="-conf slagmdialog.hmi -mainModule activate.xml
-entry cmc1"
```

## User interface

The following entries in the systemconfiguration.ini file enable the use of Easy XML for the function described above.

```
AREA0111= name:=CustomXML, dialog:=SlEECustomDialog,
panel:=SlHdStdHeaderPanel
DLG056= name:=SlEECustomDialog,
implementation:=slagmdialog.SlEECustomDialog, process:=SlHmiHost1,
preload:=false, terminate:=true, cmdline:="-conf slagmdialog.hmi"
```

## 1.15 Assigning user dialogs to reserved softkeys

A softkey is reserved in all standard operating areas for expansion of the existing functionality. Corresponding files are stored in the following directory in the system for activation and linking of an OEM softkey with a user project:

/siemens/sinumerik/hmi/template/cfg

These files contain operating area-specific XML descriptions which are interpreted and integrated in the menu tree of the operating area upon start-up of the operating software.

For integration of a separate application, the file relevant for the corresponding operating area is to be copied from the template directory into the following directory and modified:

/oem/sinumerik/hmi/template/cfg

## 1.15.1 Defining language-neutral softkey text

The TEXTFILE tag contains the file name of the text file which is to be assigned to the OEM area. The file name is to be specified without language and file extension.

### Syntax

```
<TEXTFILE>oemtextfile</TEXTFILE>
```

### Example

**oem_xml_screens_deu.ts**

```
<TEXTFILE>oem_xml_screens</TEXTFILE>
```

The softkey text to be displayed is managed in the **property** tag in the **softkey** tag. The OEM value is to be replaced by the desired text ID. The **translationContext** property references an area in the text file to which the text is assigned. In the case of easyXML, the EASY_XML context is to be specified.

### Syntax

```
<PROPERTY name="textID" type="QString">OEM</PROPERTY>
<PROPERTY name="translationContext" type="QString">OEMContext</
PROPERTY>
```

### Example

```
<PROPERTY name="textID" type="QString">MY_TEXT1</PROPERTY>
<PROPERTY name="translationContext" type="QString">EASY_XML</
PROPERTY>
```

## 1.15.2 Assigning an Easy XML area

The navigation target must also be defined in the **softkey** tag. Thus, the navigation target **CustomXML** is to be entered in the **name** attribute in the **area** tag. The attributes **dialog** and **screen** are to be deleted as these parameters are defined automatically.

### Syntax

```
<NAVIGATION target="area">
<AREA name="OEMArea" dialog="OEMDialog" screen="OEMScreen"/>
</NAVIGATION>
```

### Example

```
<NAVIGATION target="area">
<AREA name="CustomXML" />
</NAVIGATION>
```

If navigation target **area** is used, the parser expects the file xmldial.xml as a start module and the **main** menu as an entry point for menu management. In order to be able to offer multiple applications at the same time, the **switchToDialog** function is to be used. The NAVIGATION tag must then be replaced with the FUNCTION tag in this regard. The **CustomXML** area, the **SlEECustomDialog** dialog, and the name of the start module and of the main menu are to be transferred to this function as call arguments.

You return to the standard user area from the easyXML area with the **switchToArea** tag.

### Syntax

```
<switchToArea name="<Area>"/>
...
<FUNCTION args="-area CustomXML -dialog SlEECustomDialog -mainModule
<Filename> -entry <Menuname>" name="switchToDialog"/>
```

## Example project

**sldiagnose_oem.xml**

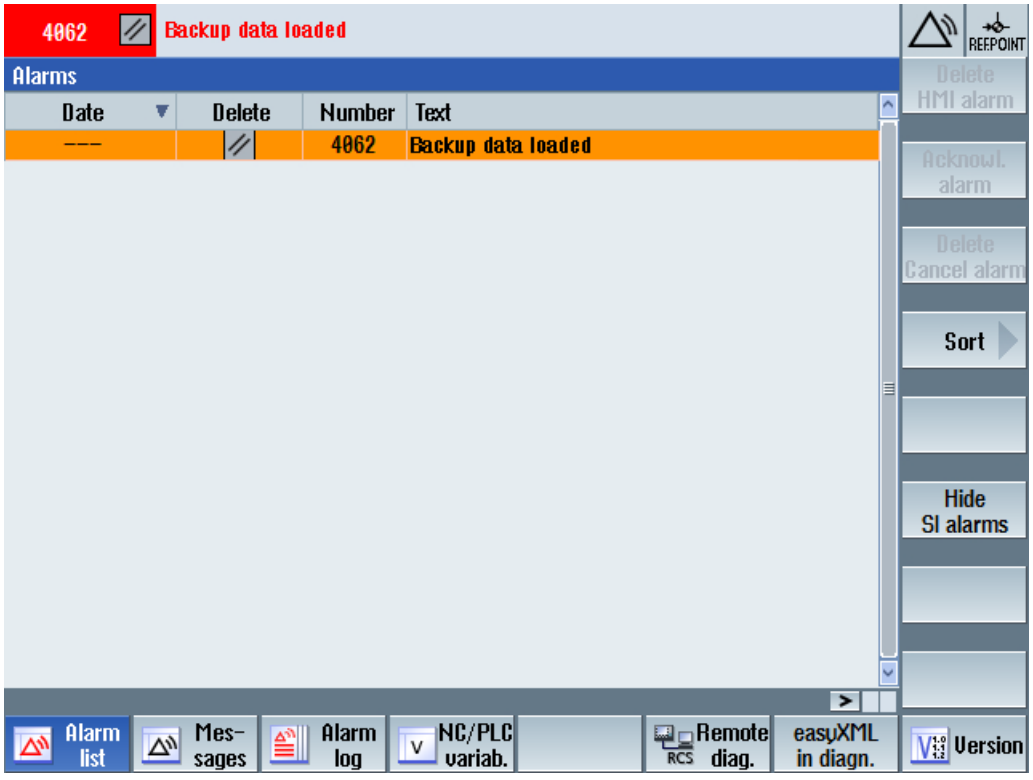
```
<!DOCTYPE DIALOG>
<DIALOG>
<TEXTFILE>oem_xml_screens</TEXTFILE>
<!-- ========================================================================
-->
<!-- OEM softkey number 7 on first horizontal main menu in area diagnosis
-->
<!-- ========================================================================
-->
<MENU name="DgGlobalHu">
 <ETCLEVEL id="0">
 <SOFTKEYGROUP name="GroupEtc">
 <SOFTKEY position="7">
  <PROPERTY name="textID" type="QString">DG_SK</PROPERTY>
  <PROPERTY name="translationContext" type="QString">EASY_XML</PROPERTY>
  <NAVIGATION target="area">
   <AREA name="CustomXML" />
  </NAVIGATION>
 </SOFTKEY>
</SOFTKEYGROUP>
</ETCLEVEL>
</MENU>
</DIALOG>
```

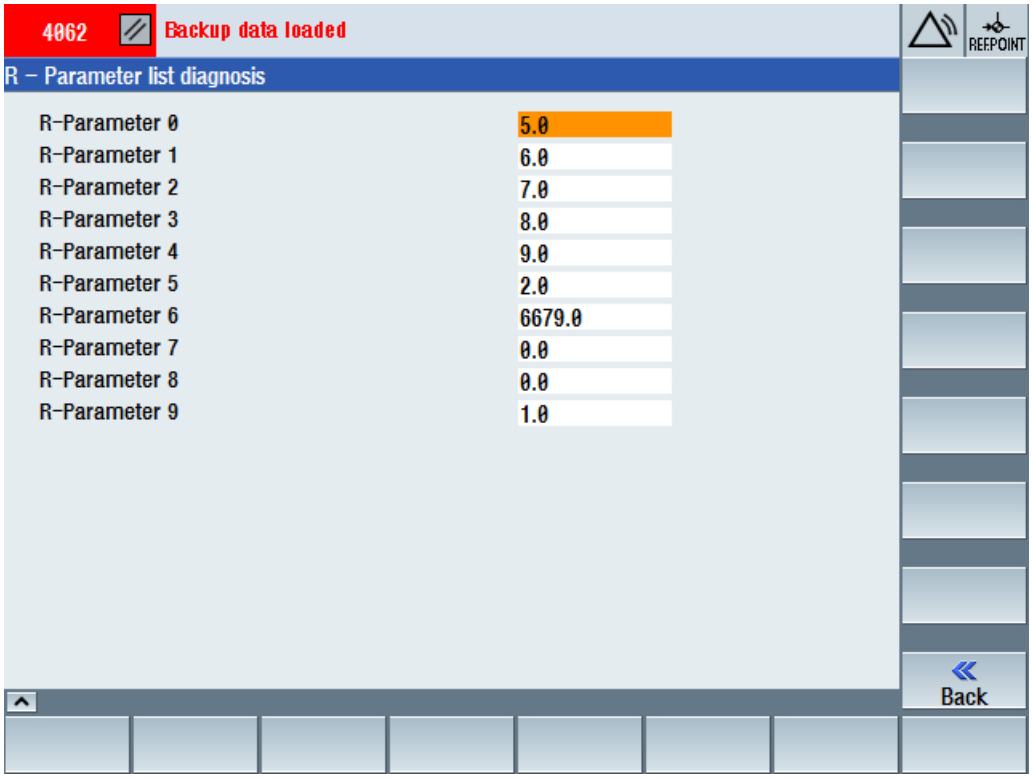
or

```
<!DOCTYPE DIALOG>
<DIALOG
<!-- ========================================================================
-->
<!-- OEM softkey number 7 on first horizontal main menu in area diagnosis
-->
<!-- ========================================================================
-->
<TEXTFILE>oem_xml_screens</TEXTFILE>
<MENU name="DgGlobalHu">
 <ETCLEVEL id="0">
 <SOFTKEYGROUP name="GroupEtc">
 <SOFTKEY position="7">
  <PROPERTY name="textID" type="QString">DG_SK</PROPERTY>
  <PROPERTY name="translationContext" type="QString">EASY_XML</PROPERTY>
  <FUNCTION args="-area CustomXML -dialog SlEECustomDialog -mainModule
dg.xml -entry main" name="switchToDialog"/>
 </SOFTKEY>
</SOFTKEYGROUP>
</ETCLEVEL>
</MENU>
</DIALOG>
```

**dg.xml**

```
<DialogGui>
<menu name = "main">
 <open_form name = "R_PARAMETER_LIST" />
 <softkey_back>
  <switchToArea name="AreaDiagnosis" dialog="SlDgDialog"/>
 </softkey_back>
</menu>
<!-- This form shows a R - parameter list -->
<form name = "R_PARAMETER_LIST">
 <init>
 <caption>R - Parameter list diagnosis</caption>
 <data_access type="true" />
  <control name = "c1" xpos = "322" ypos = "34" refvar="nck/Channel/
Parameter/R[0]" hotlink="true" />
  <control name = "c2" xpos = "322" ypos = "54" refvar="nck/Channel/
Parameter/R[1]" hotlink="true" />
  <control name = "c3" xpos = "322" ypos = "74" refvar="nck/Channel/
Parameter/R[2]" hotlink="true" />
  <control name = "c4" xpos = "322" ypos = "94" refvar="nck/Channel/
Parameter/R[3]" hotlink="true" />
  <control name = "c5" xpos = "322" ypos = "114" refvar="nck/Channel/
Parameter/R[4]" hotlink="true" />
  <control name = "c6" xpos = "322" ypos = "134" refvar="nck/Channel/
Parameter/R[5]" hotlink="true" />
  <control name = "c7" xpos = "322" ypos = "154" refvar="nck/Channel/
Parameter/R[6]" hotlink="true" />
  <control name = "c8" xpos = "322" ypos = "174" refvar="nck/Channel/
Parameter/R[7]" hotlink="true" />
  <control name = "c9" xpos = "322" ypos = "194" refvar="nck/Channel/
Parameter/R[8]" hotlink="true" />
  <control name = "c10" xpos = "322" ypos = "214" refvar="nck/Channel/
Parameter/R[9]" hotlink="true" />
 </init>
 <paint>
  <text xpos = "23" ypos = "34">R-Parameter 0</text>
  <text xpos = "23" ypos = "54">R-Parameter 1</text>
  <text xpos = "23" ypos = "74">R-Parameter 2</text>
  <text xpos = "23" ypos = "94">R-Parameter 3</text>
  <text xpos = "23" ypos = "114">R-Parameter 4</text>
  <text xpos = "23" ypos = "134">R-Parameter 5</text>
  <text xpos = "23" ypos = "154">R-Parameter 6</text>
  <text xpos = "23" ypos = "174">R-Parameter 7</text>
  <text xpos = "23" ypos = "194">R-Parameter 8</text>
  <text xpos = "23" ypos = "214">R-Parameter 9</text>
 </paint>
</form>
</DialogGui>
```


4062
Backup data loaded
REFPOINT
Alarms
Date
Delete
Number
Text
---
4062
Backup data loaded
Delete HMI alarm
Acknowl. alarm
Delete Cancel alarm
Sort
Hide SI alarms
Alarm list
Mes-sages
Alarm log
NC/PLC variab.
Remote diag.
easyXML in diagn.
Version


4062
Backup data loaded
REFPOINT
R – Parameter list diagnosis
R-Parameter 0 5.0
R-Parameter 1 6.0
R-Parameter 2 7.0
R-Parameter 3 8.0
R-Parameter 4 9.0
R-Parameter 5 2.0
R-Parameter 6 6679.0
R-Parameter 7 0.0
R-Parameter 8 0.0
R-Parameter 9 1.0
Back

## 1.15.3 Tag descriptions

### Tag softkey

Outside of the **softkey** tag, the softkey state can be set with the **state** attribute using the POSITION attribute. The softkey number for which the state should be set is to be specified as the attribute value.

### Example

```
<menu name = "menu_sktest">

 <state type="$$$define_sk_type" POSITION="2"/>

<softkey POSITION="2" type="user_controlled" picture="$$$bmp_name">
 <caption>Toggle%nSK</caption>
 <if>
 <condition>sk_type == 0 </condition>
 <then>
  <op> sk_type = 1 </op>
  <op> define_sk_type = _T"PRESSED" </op>
  <op>bmp_name = _T"f:\appl\red_led_on.bmp"</op>
  <state type="$$$define_sk_type" />
 </then>
 <else>
  <op> define_sk_type = _T"NOTPRESSED" </op>
  <op>bmp_name = _T"f:\appl\red_led_off.bmp"</op>
  <op> sk_type = 0 </op>
  <state type="$$$define_sk_type" />
 </else>
 </if>
  <print text="%s">sk_type_name</print>
  <navigation>menu_sktest</navigation>
</softkey>
...
...
```

### Tag typedef

**Note**

Pre-allocation of the elements in the examples must be removed.

Within the type definition, a variable is declared with the **element** tag. The attributes of the tag correspond with the attributes of the **let** instruction. The elements can be preallocated when creating the variables.

### Example

**RECT:**

```
<typedef name="StructRect" type="struct" >
<element name="left" type="int" />
<element name="top" type="int" />
<element name="right" type="int" />
<element name="bottom" type="int" />
</typedef>
```

**POINT:**

```
<typedef name="StructPoint" type="struct" >
<element name="x" type="int" />
<element name="y" type="int" />
</typedef>
```

**SIZE:**

```
<typedef name="StructSize" type="struct" >
<element name="width" type="int" />
<element name="height" type="int" />
</typedef>
```

## Tag let

The number of field elements is to be specified with the **dim** attribute. For a two-dimensional field, the second dimension is specified after the first dimension separated by a comma. A field element is accessed via the field index, which is specified in square brackets after the variable name. Dimensions of the arrays can be defined directly or via the content of a variable.

## Example

```
 <let name="d1" >3</let>

<let name="list_string" dim="3" type="string" />
```

or

```
<let name="list_string" dim="$d1" type="string" />
...
...
<op>
  list_string[2] = _T"test";
</op>
```

or

```
<let name="d1" >3</let>
<let name="d2" >6</let>

<let name="list_string3" dim="3, $d2" type="string" />

<op>
  list_string3[2, 5] = _T"test";
</op>
```

# 1.16 Creating language-neutral texts

All texts used in the dialog forms can be managed language-neutral, where text identifiers are used in the dialogs which are replaced with text in the currently selected language during runtime. This text is to be saved together with the text identifier for every available language in a text file in UTF8 format.

The file oem_xml_screens_<language code>.ts is assigned to the Easy XML function as a text file by default. The EASY_XML identifier is to be specified as text context.

Two dollar symbols are to be prefixed to the text identifier in the script to mark a test identifier.

## Structure

**oem_xml_screens_deu.ts**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
<context>
    <name>EASY_XML</name>
    <message>
    <source>MY_TEXT1</source>
    <translation>text1 from textfile</translation>
    </message>
    <message>
    <source>MY_TEXT2</source>
    <translation>text2 from textfile</translation>
    </message>
</context>
</TS>
```

## Example

```
<text xpos ="120" ypos="158">$$MY_TEXT1</text>
```

# 1.17 Project-specific text files

## 1.17.1 Creating project-specific text files

For managing separate projects, separate text files can be used for a project, each form, and each menu. The TEXTFILE attribute gives notification in the corresponding tags.

The name of the text file is to be transferred as an attribute value without language identifier and file extension. The texts can continue to be accessed until the form or the menu is closed. If a text file is loaded with the tag **DialogGui**, the content may be accessed until the project is exited. The search for the text identifier begins in the file last loaded.

### Example

**main_form_screens_deu.ts**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
<context>
    <name>EASY_XML</name>
    <message>
    <source>MAIN_FORM_TITLE</source>
    <translation>user text file title</translation>
    </message>
</context>
</TS>
```

### Activation

```
<DialogGui textfile="main_form_screens">
...
...
</DialogGui>
```

or

```
<form name="main_form" textfile="main_form_screens">
...
...
</form>
```

or

```
<menu name="main" textfile="main_form_screens">
...
...
</menu>
```

## Example dialog

**main_form_screens_deu.ts**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
<context>
    <name>EASY_XML</name>
    <message>
    <source>MAIN_FORM_TITLE</source>
    <translation>user text file title</translation>
    </message>
    <message>
    <source>MAIN_FORM_TEXT</source>
    <translation>text from text file</translation>
    </message>
</context>
</TS>
```

**main2_sktext_screens_deu.ts**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
<context>
    <name>EASY_XML</name>
    <message>
    <source>SK1</source>
    <translation>Softkey1</translation>
    </message>
</context>
</TS>
```

**form2_screens_deu.ts**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
<context>
    <name>EASY_XML</name>
    <message>
    <source>FORM2_TITLE</source>
    <translation>Form2 Title</translation>
    </message>
    <message>
    <source>FORM2_TEXT</source>
    <translation>Form2 text from text file</translation>
    </message>
</context>
</TS>
```

### Dialog script

**xmldial.xml**

```
<DialogGui textfile="main_form_screens">
<menu name = "main">
    <open_form name = "main_form" />
    <softkey POSITION="1">
    <caption>Menu 2</caption>
    <navigation>menu_2</navigation>
    </softkey>
</menu>
<menu name = "menu_2" textfile="main2_sktext_screens">
    <open_form name = "form2" />
    <softkey POSITION="1">
    <caption>$$SK1</caption>
    </softkey>
    <softkey_back>
    <navigation>main</navigation>
    </softkey_back>
</menu>
<form name="main_form" >
 <init>
    <data_access type = "true" />
    <caption>$$MAIN_FORM_TITLE</caption>
 </init>
 <paint>
    <text xpos="5" ypos="30">$$MAIN_FORM_TEXT</text>
 </paint>
</form>
<form name="form2" textfile="form2_screens">
 <init>
    <data_access type = "true" />
    <caption>$$FORM2_TITLE</caption>
 </init>
 <paint>
    <text xpos="5" ypos="30">$$FORM2_TEXT</text>
 </paint>
</form>
</DialogGui>
```

## 1.17.2 Specifying text context

Grouping of the texts is possible within a text file by way of independently named area names. The area identifier is defined within the **context** tag with the **name** tag.

### Syntax

```
<context>
 <name>name</name>
</context>
```

## Example

```
<context>
 <name>my context 1</name>
...
...
</context>
<context>
 <name>my context 2</name>
...
...
</context>
```

If particular context is being used, the context name is to be specified together with the text file name for a project, a form or a menu via the TEXTCONTEXT attribute. Access to texts stored within the context remains possible until a new context is set. Thus, a context set for the project is replaced by the context set for a form or a menu.

## Project example

**form2_screens_deu.ts**

```
?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
<context>
 <name>EASY_XML</name>
  <message>
   <source>FORM2_TITLE</source>
   <translation>Form2 Title</translation>
  </message>
  <message>
   <source>FORM2_TEXT</source>
   <translation>Form2 text from text file</translation>
  </message>
</context>
<context>
 <name>MY_FORM_CONTEXT</name>
  <message>
   <source>FORM3_TITLE</source>
   <translation>Title from the text context MY_FORM_CONTEXT</translation>
  </message>
  <message>
   <source>FORM3_TEXT</source>
   <translation>Form3 text from text file</translation>
  </message>
</context>
<context>
 <name>MY_SOFTKEY_CONTEXT</name>
  <message>
   <source>MENU3_SK1</source>
   <translation>SK%n1</translation>
  </message>
</context>
</TS>
```

**using_textcontext.xml**

```
...
<menu name = "menu3" textfile="form2_screens"
textcontext="MY_SOFTKEY_CONTEXT">
 <open_form name = "menu3_form" />
 <softkey POSITION="1">
  <caption>$$SK1</caption>
 </softkey>
 <softkey_back>
  <navigation>main</navigation>
 </softkey_back>
</menu>
<form name="menu3_form" textfile="form2_screens"
textcontext="MY_FORM_CONTEXT">
 <init>
  <data_access type = "true" />
  <caption>$$FORM3_TITLE</caption>
 </init>

 <paint>
  <text xpos="5" ypos="30">$$FORM3_TEXT</text>
 </paint>
</form>
...
...
```

# Generating commissioning dialogs

# 2

## 2.1 Overview of functions

### Purpose

The "Easy Extend" function provides a simple way of commissioning, activating, deactivating, or testing optional equipment. The available equipment and device states are displayed in a list by the control system. The system can manage a maximum of 64 devices.

Softkeys are used to activate or deactivate a device.

The "Easy Extend" function is available in the "Parameters" operating area.

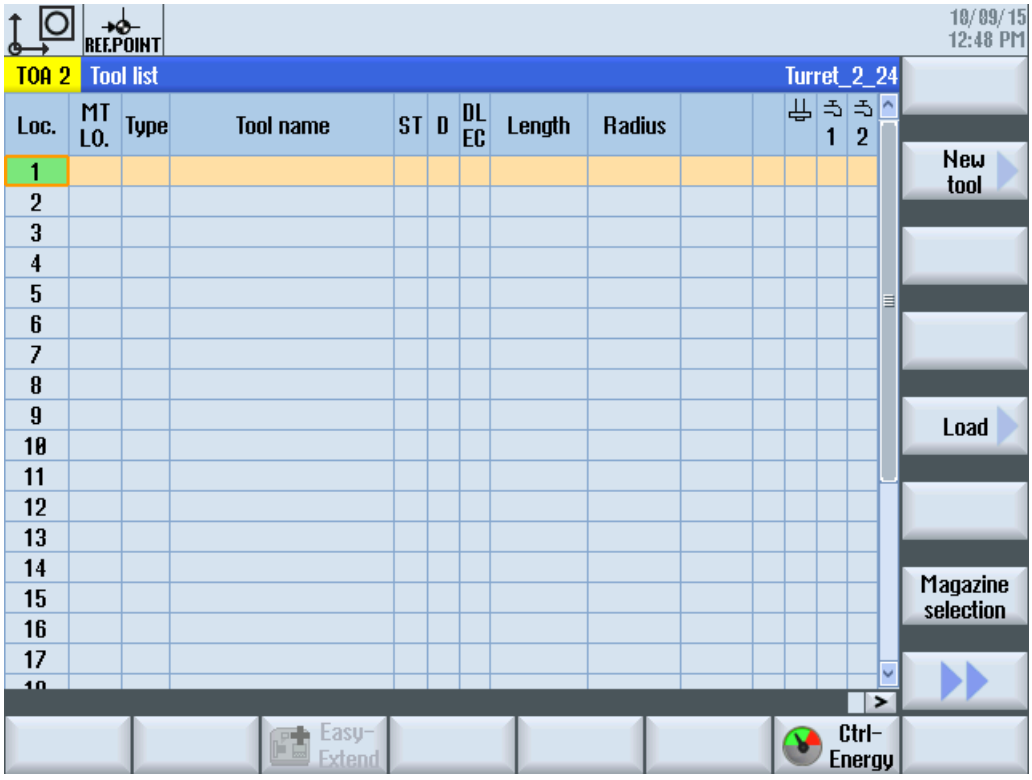



Figure 2-1 Main screen "Parameters"

## Configuration

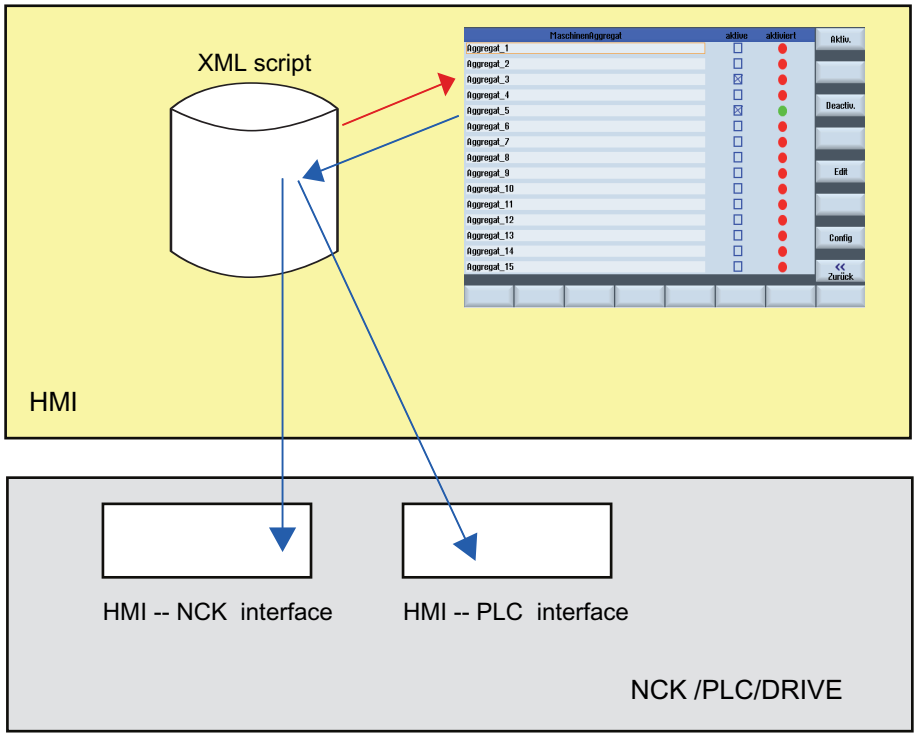



Figure 2-2 How "Easy Extend" works

To use the "Easy Extend" function, the following functions must be configured by the machine manufacturer:

- **PLC ↔ HMI interface**
  The optional devices are managed via the interface between the user interface and the PLC.
- **Script processing**
  The machine manufacturer saves the sequences to be executed for commissioning, activating, deactivating or testing a device, in a statement script.
- **Parameter dialog (optional)**
  The parameter dialog shows device information that is saved in the script file.

## Storage of the files

The files belonging to "Easy Extend" are stored on the system CompactFlash card in directory "dvm" (machine manufacturer).

| File | Name | Target directory |
|---|---|---|
| Text file | oem_aggregate_xxx.ts | \oem\sinumerik\hmi\lng |
| Script file | agm.xml | \oem\sinumerik\hmi\dvm |
| Archive file | Any | \oem\sinumerik\hmi\dvm\archives |
| PLC user program | Any | PLC |

# 2.2 Configuration in the PLC user program

## Loading configurations

The configurations created are transferred to the manufacturer directory of the control, with the script and text file. Additionally, the corresponding PLC user program should be loaded.

## Programming the equipment

Communication between the operator component and the PLC takes place in the PLC user program via a data block defined by the programmer, in which 128 words are reserved for the management of the devices. The description for the data block identifier is provided in Chapter "PLC_INTERFACE (Page 219)."

The data block must be generated and loaded as a user block. The relevant block must be declared to the EasyExtend function using the **plc_interface** tag in the "agm.xml" script.

**Note**

**Compatibility**

We recommend defining data block DB9905 as the PLC interface so that the scripts are also compatible with SINUMERIK 828D.

**Example:**

```
<plc_interface name = "plc/db1000.dbb0" />
```

To create the block, data formats and symbolic identifiers from the following table can be used:

| Data format/symbolic identifier | Meaning |
|---|---|
| DBX0.0 Enable_1 BOOL Bit OFF OFF HMI → PLC | Device has been started up |
| DBX0.1 Activate_1 BOOL Bit OFF OFF HMI → PLC | Device is to be activated |
| DBX0.2 Deactivate_1 BOOL Bit OFF OFF HMI → PLC | Device is to be deactivated |
| DBB1 Res_1 BYTE Unsigned 0 0 | Reserved for future use |
| DBX2.0 IsActive_1 BOOL Bit OFF OFF PLC → HMI | Device is active |
| DBX2.1 Error_1 BOOL Bit OFF OFF PLC → HMI | Device has an error |
| DBB3 DeviceId_1 BYTE Unsigned 0 0 | Unique device number |

PLC words are assigned beginning with Device 1:

| Data block | Device designation |
|---|---|
| DB9905.DBB0 | Device 1 |
| DB9905.DBB4 | Device 2 |
| ... | ... |
| DB9905.DBB192 | Device 49 |
| DB9905.DBB196 | Device 50 |

Four bytes with the following meanings are used for each device:

| Byte | Bit | Description | |
|---|---|---|---|
| 0 | 0 | == 1 | Device has been started up (HMI acknowledgment) |
| | 1 | == 1 | Device is to be activated (HMI request) |
| | 2 | == 1 | Device is to be deactivated (HMI request) |
| | 3-7 | Reserved | |
| 1 | 0-7 | Reserved | |
| 2 | 0 | == 1 | Device is active (PLC acknowledgment) |
| | 1 | == 1 | Device has an error |
| | 2-7 | Reserved | |
| 3 | 0-7 | Unique identifier for the device | |

## Storage of the files

The Easy Extend files are stored on the system CompactFlash card in the "oem" (MANUFACTURER) and "oem_i" (INDIVIDUAL) directory.

**Note**

The file name may only contain lower case letters.

| File | Name | Target directory |
|---|---|---|
| Text file | oem_aggregate_xxx.ts | /oem/sinumerik/hmi/lng/<br>/oem_i/sinumerik/hmi/lng/ |
| Script file | agm.xml | /oem/sinumerik/hmi/dvm<br>/oem_i/sinumerik/hmi/dvm |
| Archive file | Any | /oem/sinumerik/hmi/dvm/archives<br>/oem_i/sinumerik/hmi/dvm/archives |
| PLC user program | Any | PLC |

## General sequence

The machine manufacturer must execute the following steps to make the required data available:

1. Creating a PLC user program which activates the device during activation on the PLC.
2. Commissioning of the "standard machine" followed by backup of the data in a series startup archive.
3. Installation of the devices, commissioning, followed by read-out of the data as a differential series startup archive.

**Note**

**Changing the machine configuration**

Should there be any need to edit the drive machine data, this should be adapted in the control first. This procedure should be repeated for all devices and constellations.

## Adding axes

If the machine is extended with machine axes, it is important to install the drive objects (DO) in a fixed sequence because the series startup archive contains the constellation of the machine manufacturer's reference machine and cannot be applied if the sequence is changed.

It is recommended that the following settings be selected for the "control components":

- NC data
- PLC data
- Drive data
  - ACX format (binary)

# 2.3 Display on the user interface

## Dialogs on the user interface

The following dialogs are available for the "Easy Extend" function:

- The control offers a **configurable dialog**, in which the available devices are shown.
- If first commissioning has not taken place yet, the control opens the **commissioning dialog**.

If a commissioning procedure (XML instruction: "START_UP") has been programmed, and the device has still not been commissioned, then the control starts the commissioning procedure.

This involves a complete data backup before the series startup archives saved in the script file are read in.

In the event of an error, the commissioning engineer can decide whether to roll back the commissioning procedure or to rectify possible errors in machine configurations manually.

- Commissioning can be aborted early with the "Cancel" function. The control then copies the previously saved commissioning files back.

If the machine has to be switched off after successful completion of the commissioning, the XML statement "POWER_OFF" can be used to program that a corresponding message is output on the control.

# 2.4 Creating language-dependent texts

## Structure of text file

The XML files with the language-dependent texts must be created in UTF8 format:

## Examples

**oem_aggregate_eng.ts**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
<context>
    <name>EASY_EXTEND</name>
    <message>
    <source>DEVICE_ONE</source>
    <translation>Device one</translation>
    </message>
    <message>
    <source>DEVICE_TWO</source>
    <translation>Device two</translation>
    </message>
</context>
</TS>
```

**oem_aggregate_deu.ts**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
<context>
    <name>EASY_EXTEND</name>
    <message>
    <source>DEVICE_ONE</source>
    <translation>First device</translation>
    </message>
    <source>DEVICE_TWO</source>
    <translation>Second device</translation>
    </message>
</context>
</TS>
```

## 2.5 User example for a power unit

### Activating the drive object

The drive object to be activated has already been commissioned and deactivated again by the machine manufacturer, to market the axis (axes) as an option.

To activate the axis carry out the following steps:

- Activate the drive object via p0105.
- Enable 2nd axis in the channel machine data.
- Back up the drive machine data via p0971.
- Wait until the data has been written.
- Restart the NCK and the drives.

**Programming:**

```
<DEVICE>
  <list_id>1</list_id>
  <name> "Activate the drive" </name>

  <SET_ACTIVE>
    <data name = "drive/dc/p105[DO5]">1</data>
    <data name = "$MC_AXCONF_MACHAX_USED[4]">5</data>
    <data name = "drive/dc/p971[DO5]">1</data>
    <while>
      <condition> "drive/dc/p971[DO5]" !=0 </condition>
    </while>
    <control_reset resetnc ="true" resetdrive = "true"/>
  </SET_ACTIVE>

  <SET_INACTIVE>
    <data name = "drive/dc/p105[DO5]">0</data>
    <data name = "$MC_AXCONF_MACHAX_USED[4]">0</data>
    <data name = "drive/dc/p971[DO5]">1</data>
    <while>
      <condition> "drive/dc/p971[DO5]" !=0 </condition>
    </while>
    <control_reset resetnc ="true" resetdrive = "true"/>
  </SET_INACTIVE>

</DEVICE>
```

### Activating the PLC-controlled device

The device is addressed via output byte 10 and signals data set ready to the PLC via input byte 9.

The output byte is set to the specified coding for activation. The While loop then waits for the data set ready of the device.

**Programming:**

```
<SET_ACTIVE>
  <DATA name = "plc/qb10"> 8 </DATA>
  <while>
    <condition> "plc/ib9" !=1 </condition>
  </while>
</SET_ACTIVE>
```

# 2.6 Script language

**Note**

All script elements described in the Generating user dialogs (Page 5) function form the basis for the "Easy Extend" function. Additional script elements are defined to manage additional devices.

## Program parts of the script

The script is divided into the following areas:

- "Easy Extend" identifier
- Frame to define the actions that can be executed for a device
- Identifier for the device
- Identifier for commissioning the device
- Identifier for activating the device
- Identifier for deactivating the device
- Identifier for testing the device
- Identifier for the parameter dialog

The individual tags are described in the following chapters.

## Description

| Identifier <tag> | Meaning |
|---|---|
| AGM | Identifier for the "Startup Wizard" |
| DEVICE | Identifier for the description of the device.<br><br>**Attributes:**<br>• **option_bit**<br>The device is assigned a fixed bit number for the option management. |
| NAME | The identifier specifies the name of the device to be displayed in the dialog.<br>If a text reference is used, the dialog displays the text which is saved for the identifier. |
| START_UP | The identifier contains a description of the sequences required for commissioning the device. |

| Identifier <tag> | Meaning |
|---|---|
| SET_ACTIVE | The identifier contains a description of the sequences required to activate the device.<br>**Attributes:**<br>• **timeout**<br>The attribute permits a timeout to be specified in seconds. The system interrupts processing if the script has still not been completed after this time. |
| SET_INACTIVE | The identifier contains a description of the sequences required to shut down the device.<br>**Attributes:**<br>• **timeout**<br>The attribute permits a timeout to be specified in seconds. The system interrupts processing if the script has still not been completed after this time. |
| TEST | The identifier contains the statements for testing the operating capability of a device.<br>**Attributes:**<br>• **timeout**<br>The attribute permits a timeout to be specified in seconds. The system interrupts processing if the script has still not been completed after this time. |
| UID | Unique numerical identifier to identify the device in the PLC ↔ HMI interface. |
| VERSION | Identifier for a version |

## Negative acknowledgment of the function execution

With the automatically provided variable "$actionresult", the system can inform the XML parser of a negative execution result. If the value is set to zero, the parser aborts the function processing.

## Example

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE AGM>
<AGM>                              Identifier for the "Startup Wizard"
<DEVICE>
  <NAME> Device 1 </NAME>          Identifier for the device
  <START_UP>                       Identifier for commissioning the device
  …
  </START_UP>
  <SET_ACTIVE>                     Identifier for activating the device
  …
  </SET_ACTIVE>
  <SET_INACTIVE>                   Identifier for deactivating the device
  …
  </SET_INACTIVE>
```

```
  <TEST>                          Identifier for testing the device
  …
  </TEST>
</DEVICE>
…
</AGM>
```

## 2.6.1 CONTROL_RESET

### Description

This identifier allows one or more control components to be restarted. Execution of the script is only continued when the control has resumed cyclic operation.

### Programming

| | | |
|---|---|---|
| Identifier: | **CONTROL_RESET** | |
| Syntax: | <CONTROL_RESET resetnc="TRUE" /> | |
| Attributes: | resetnc="true" | The NC component is restarted. |
| | resetdrive="true" | The drive components are restarted. |

## 2.6.2 FILE

### Description

The identifier enables standard archives to be read in or created.

- Reading in an archive:
  The file name of the archive must be specified for reading in an archive.
- Creating an archive:
  If the attribute create= "true" is specified, the function creates a standard archive (*.arc) under the specified name and stores the file in the …/dvm/archives directory.

### Programming

| | | |
|---|---|---|
| Identifier: | **FILE** | |
| Syntax: | <file name ="<archive name>" /> | |
| | <file name ="<archive name>" create="true" class="<data classes>" group="<area>" /> | |
| Attributes: | name | Identifier for the file name |

| | |
|---|---|
| create | A commissioning archive is created under the specified name in the …/dvm/archives/ directory. |
| group | Specifies the data groups that are to be contained in the archive. If several data groups are to be saved, the groups should be separated by a blank.<br>The following data groups can be contained in the archive:<br>• NC<br>• PLC<br>• HMI<br>• DRIVES |

### Example

```
<!-- Create data class archive -->
<file name="user.arc" create="true"
group="nc plc hmi" />

<!—Read archive into the control->
<file name="user.arc" />
```

## 2.6.3 OPTION_MD

### Description

The identifier allows option machine data to be redefined. As delivered, the system uses MD14510 $MN_USER_DATA_INT[0] to $MN_USER_DATA_INT[3].

If the PLC user program manages the options, the appropriate data words must be provided in a data block or GUD.

The data is structured in bits. Starting with bit 0, there is a fixed assignment of the bits to the listed devices, i.e. bit 0 is assigned to device 1, bit 1 to device 2, etc. If more than 16 devices are managed, the address identifiers of the device groups 1-3 are assigned via the area index.

**Note**

**Converting the value range**

The value range of MD14510 $MN_USER_DATA_INT[i] is from -32768 to +32767. To activate the devices bit-by-bit via the machine data dialog, the bit combination must be converted to decimal representation.

### Programming

| | | |
|---|---|---|
| Identifier: | **OPTION_MD** | |
| Syntax: | Area 0: | |
| | <option_md name = "Address identifier of the data" /> | |
| | OR: | |
| | <option_md name = "Address identifier of the data" index= "0"/> | |
| | Area 1 to 3: | |
| | <option_md name = "Address identifier of the data" index= "Area index"/> | |
| Attributes: | name | Identifier for the address, e.g. $MN_USER_DATA_INT[0] |
| | index | Identifier for the area index:<br>0 (default setting): Device 1 to 16<br>1: Device 17 to 32<br>2: Device 33 to 48<br>3: Device 49 to 64 |

## 2.6.4 PLC_INTERFACE

### Description

This identifier permits the PLC ↔ HMI interface to be redefined. The system expects 128 addressable words.

**Note**

The identifier is not preset and must be defined by the programmer.

### Programming

| | | |
|---|---|---|
| Identifier: | **PLC_INTERFACE** | |
| Syntax: | <plc_interface name = "Address identifier of the data" /> | |
| Attributes: | name | Identifier for the address, e.g. "plc/mb170" |

**Example:** `plc/mb170`

## 2.6.5 POWER_OFF

### Description

Identifier for a message prompting the operator to switch the machine off. The message text is permanently saved in the system.

### Programming

| | |
|---|---|
| Identifier: | **POWER_OFF** |
| Syntax: | <power_off /> |
| Attributes: | -- |

## 2.6.6 WAITING

### Description

After a reset of the NC or the drive, there is a wait for the restart of the respective component.

### Programming

| | | |
|---|---|---|
| Identifier: | **WAITING** | |
| Syntax: | <WAITING WAITINGFORNC ="TRUE" /> | |
| Attributes: | waitingfornc="true" | There is a wait for the restart of the NC. |
| | waitingfordrive="true" | There is a wait for the restart of the drive. |

## 2.6.7 XML identifiers for the dialog

### Dialog for the parameterization

A dialog can be configured for each device so that additional parameters can be set or output during runtime. This is displayed by pressing the "Additional parameters" softkey.

All of the script elements described in Chapter Generating user dialogs (Page 5) can be used to generate the dialog.

### Example

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE AGM>
<AGM>
<DEVICE>
  <NAME> Device 1 </NAME>
  <START_UP>
  …
  </START_UP>
  <SET_ACTIVE>
  …
  </SET_ACTIVE>
…
  <FORM name="<dialog name>">          Identifier for a user dialog
    <INIT>
      <CONTROL name = "edit1" .../>    Identifier for an input field
    </INIT>
    <PAINT>                            Identifier for text or image display
      <TEXT>hello world !</TEXT>
    </PAINT>
  </FORM>

</DEVICE>
…
</AGM>
```

## 2.6.8 SOFTKEY_OK, SOFTKEY_CANCEL

### Description

The identifier SOFTKEY_OK overwrites the standard behavior when closing a dialog by means of the "OK" softkey. The identifier SOFTKEY_CANCEL overwrites the standard behavior when closing a dialog by means of the "CANCEL" softkey.

The following functions can be performed within this identifier:

- Data manipulation
- Conditional processing
- Loop processing

## Programming

| | |
|---|---|
| Identifier: | **SOFTKEY_OK** |
| Syntax: | <SOFTKEY_OK><br>…<br></SOFTKEY_OK> |
| Identifier: | **SOFTKEY_CANCEL** |
| Syntax: | <SOFTKEY_CANCEL><br>…<br></SOFTKEY_CANCEL> |

# Index


## C

Configuration file, 7

## E

Easy Extend, 205
Easy XML diagnostics, 21

## F

Finger gestures, 153

## G

Generating commissioning dialogs, 205

## M

Menu tree, 7

## S

Start softkey, 7

## X

XML
- Operators, 49
- Syntax, 48
- System variables, 50

XML diagnostics, 21
XML functions
- Abs, 143
- AddItem, 135
- ARCOS, 127
- ARCSIN, 127
- ARCTAN, 128
- Bitmap dimensions, 142
- CEIL, 143
- Control form color, 117
- Control local time, 118
- Copy value and read, 107
- Copying and writing a value, 108
- Cosine, 126
- Delete control, 134
- Delete number of characters of a string, 124
- DeleteItem, 136
- Deleting a file, 131
- display resolution, 111
- Empty, 138
- Exist, 133
- Extracting script parts, 132
- Find substring, 124
- Find substring (reverse), 125
- FLOOR, 143
- Get cursor selection, 139
- getfocus, 138
- GetItem, 140
- GetItemData, 140
- imagebox, 156
- ImageBoxGet, 142
- ImageBoxSet, 69, 141, 157
- InsertItem, 136
- LoadItem, 137
- LOG, 144
- LOG10, 144
- MAX, 144
- MIN, 144
- MMC, 145
- NC program selection, 133
- Ncfunc bico to int, 116
- Ncfunc Get drive by axis index, 113
- Ncfunc Get drive by axis name, 112
- Ncfunc Get drive by bud address, 115
- Ncfunc Get drive by drive name, 114
- Ncfunc int to bico, 116
- Ncfunc is bico str valid, 116
- Ncfunc password, 117
- PI service, 109, 110
- PI service, channel-related:, 111
- POW, 144
- RANDOM, 144
- Reading a file, 129
- ROUND, 143
- Screen resolution, 142
- Screen resolution HMI, 142
- SDEG, 143
- Set cursor selection, 139
- setfocus, 139
- Setting an individual bit, 134
- Sine, 126
- SQRT, 143
- SRAD, 143

String comparison, 119, 120
String find, 150
String GetAt, 151
String reverse find, 151
String SetAt, 152
String Split, 152
string.icmp, 120
string.insert, 123
string.left, 120
string.length, 122
string.middle, 121
string.remove, 123
string.replace, 122
string.right, 121
Tangent, 127
Title bar height, 143
Writing to a file, 130
XML identifier
AGM, 214
AUTOSCALE_CONTENT, 52
BOX, 86
BREAK, 24
CAPTION, 61, 163, 176
CLOSE, 61
CLOSE_FORM, 62
CONTROL, 24, 63, 167, 173
CONTROL_RESET, 24, 217
CREATE_CYCLE, 92
CREATE_CYCLE_EVENT, 25
DATA, 26
DATA_ACCESS, 71
DATA_LIST, 27
DEVICE, 214
DO_WHILE, 27
DYNAMIC_INCLUDE, 28
EDIT_CHANGED, 71
ELSE, 28
FILE, 217
FOCUS_IN, 61
FOR, 29
FORM, 29, 52
FUNCTION, 86, 171
FUNCTION_BODY, 87
GESTURE_EVENT, 71, 155
HELP_CONTEXT, 70
HMI_RESET, 29
IF, 30
IMG, 83, 84, 85
INCLUDE, 30
INDEX_CHANGED, 71
INIT, 54
KEY_EVENT, 55
LET, 31, 198
LOCK_OPERATING_AREA, 34
MENU, 72
MESSAGE, 57, 159
MOUSE_EVENT, 56
MSG, 34
MSGBOX, 34
NAME, 214
NAVIGATION, 72
NC_INSTRUCTION, 91
OP, 35
OPEN_FORM, 73
OPERATION, 36
OPTION_MD, 219
PAINT, 61
PASSWORD, 36
PLC_INTERFACE, 219
POWER_OFF, 36, 220
PRINT, 37
PROGRESS_BAR, 38
PROPERTY, 74, 75, 76, 77, 78, 161, 165, 172
RECALL, 89
REQUEST, 88
RESIZE, 59
SEND_MESSAGE, 39, 58
SET_ACTIVE, 215
SET_INACTIVE, 215
SHOW_CONTROL, 39
SLEEP, 40
SOFTKEY, 78, 197
SOFTKEY_ACCEPT, 82
SOFTKEY_BACK, 82
SOFTKEY_CANCEL, 81, 222
SOFTKEY_OK, 81, 222
START_UP, 214
STOP, 40
SWITCH, 40
SWITCHTOAREA, 41
SWITCHTODYNAMICTARGET, 41
TEST, 215
TEXT, 82
THEN, 41
TIMER, 61
TYPE_CAST, 41
TYPEDEF, 42, 197
UID, 215
UNLOCK_OPERATING_AREA, 43
UPDATE_CONTROLS, 89
VERSION, 215
WAITING, 43, 220
WHILE, 44
XML_PARSER, 44

SIEMENS

# SINUMERIK

# SINUMERIK 840D sl
# SINUMERIK Integrate Run
# MyScreens (BE2)

Programming Manual


Introduction 1

Getting Started 2

Fundamentals 3

Dialogs 4

Variables 5

Programming commands 6

Graphic and logic elements 7

"Custom" operating area 8

Dialog selection 9

Examples for cycle masks 10

Configuration in the side screen 11

Configuration in the Display Manager 12

Reference lists A

Tips and tricks B

Animated elements C


Valid for

CNC software Version 4.8 SP3


08/2018
6FC5397-1DP40-6BA2

## 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

Document order number: 6FC5397-1DP40-6BA2
℗ 08/2018 Subject to change

Copyright © Siemens AG 2018.
All rights reserved

# Table of contents


**1 Introduction....9**

**2 Getting Started....13**

2.1 Introduction....13

2.2 Project example....14
2.2.1 Task description....14
2.2.2 Creating the configuration file (example)....18
2.2.3 Saving the configuration file in the OEM directory (example)....21
2.2.4 Creating the online help (example)....22
2.2.5 Integrating the online help and saving the files to the OEM directory (example)....25
2.2.6 Copying "easyscreen.ini" into the OEM directory (example)....26
2.2.7 Registering the COM file in "easyscreen.ini" (example)....26
2.2.8 Testing the project (example)....26

**3 Fundamentals....29**

3.1 Structure of configuration file....29

3.2 Structure of the menu tree....31

3.3 Definition and functions for start softkeys....33
3.3.1 Defining the start softkey....33
3.3.2 Functions for start softkeys....34

3.4 Troubleshooting (log book)....36

3.5 Notes on "easyscreen.ini"....38

3.6 Notes for personnel changing over to "Run MyScreens"....41

3.7 Extended configuration syntax....43

3.8 SmartOperation and MultiTouch operation....45

**4 Dialogs....47**

4.1 Structure and elements of a dialog....47
4.1.1 Defining a dialog....47
4.1.2 Defining dialog properties....49
4.1.3 Defining dialog elements....56
4.1.4 Defining dialogs with multiple columns....58
4.1.5 Password dialogs....59
4.1.6 Using display images/graphics....61

4.2 Defining softkey menus....62
4.2.1 Changing softkey properties during runtime....64
4.2.2 Language-dependent text....67

4.3 Configuring the online help....71
4.3.1 Overview....71
4.3.2 Generating HTML files....72
4.3.3 Generating the help book....75

4.3.4 Integrating the online help in SINUMERIK Operate....77
4.3.5 Saving help files....79
4.3.6 Help files in PDF format....79
5 Variables....81
5.1 Defining variables....81
5.2 Application examples....82
5.3 Example 1: Assigning the variable type, texts, help display, colors, tooltips....84
5.4 Example 2: Assigning the Variable Type, Limits, Attributes, Short Text Position properties....85
5.5 Example 3: Assigning the Variable Type, Default, System or User Variable, Input/Output Field Position properties....86
5.6 Example 4: Toggle field and list field....87
5.7 Example 5: Image display....88
5.8 Example 6: Progress bar....89
5.9 Example 7: Password input mode (asterisk)....91
5.10 Variable parameters....92
5.11 Details on the variable type....98
5.12 Details on the toggle field....101
5.13 Details on the default setting....103
5.14 Details on the position of the short text, position of the input/output field....104
5.15 Use of strings....106
5.16 CURPOS variable....108
5.17 CURVER variable....109
5.18 ENTRY variable....110
5.19 ERR variable....111
5.20 FILE_ERR variable....112
5.21 FOC variable....114
5.22 Variable S_ALEVEL....115
5.23 S_CHAN variable....116
5.24 Variable S_CONTROL....117
5.25 Variable S_LANG....118
5.26 Variable S_NCCODEREADONLY....119
5.27 Variables S_RESX and S_RESY....120
6 Programming commands....121
6.1 Operators....121
6.1.1 Mathematical operators....121
6.1.2 Bit operators....124

6.2 Methods....126
6.2.1 ACCESSLEVEL....126
6.2.2 CHANGE....127
6.2.3 CHANNEL....128
6.2.4 CONTROL....129
6.2.5 FOCUS....129
6.2.6 LANGUAGE....130
6.2.7 LOAD....130
6.2.8 UNLOAD....131
6.2.9 OUTPUT....132
6.2.10 PRESS....133
6.2.11 PRESS(ENTER)....134
6.2.12 PRESS(TOGGLE)....134
6.2.13 RESOLUTION....135
6.2.14 RESUME....135
6.2.15 SUSPEND....136
6.2.16 Example: Version management with OUTPUT methods....136
6.3 Functions....138
6.3.1 Reading and writing drive parameters: RDOP, WDOP, MRDOP....138
6.3.2 Subprogram call (CALL)....140
6.3.3 Define block (//B)....141
6.3.4 Check Variable (CVAR)....142
6.3.5 CLEAR_BACKGROUND....143
6.3.6 Copy Program file function (CP)....144
6.3.7 Delete Program file function (DP)....144
6.3.8 Exist Program file function (EP)....145
6.3.9 Move Program file function (MP)....146
6.3.10 Select Program file function (SP)....147
6.3.11 File accesses: RDFILE, WRFILE, RDLINEFILE, WRLINEFILE....148
6.3.12 Dialog line (DLGL)....150
6.3.13 DEBUG....151
6.3.14 Exit dialog (EXIT)....152
6.3.15 Dynamic manipulation of the lists of toggle fields or list box fields....153
6.3.16 Evaluate (EVAL)....156
6.3.17 Exit Loading Softkey (EXITLS)....157
6.3.18 Function (FCT)....157
6.3.19 Generate code (GC)....159
6.3.20 Password functions....162
6.3.21 Load Array (LA)....163
6.3.22 Load Block (LB)....164
6.3.23 Load Mask (LM)....165
6.3.24 Load Softkey (LS)....166
6.3.25 Load Grid (LG)....167
6.3.26 Multiple selection SWITCH....168
6.3.27 Multiple Read NC PLC (MRNP)....169
6.3.28 PI services....171
6.3.29 Reading (RNP) and writing (WNP) system or user variables....172
6.3.30 RESIZE_VAR_IO and RESIZE_VAR_TXT....173
6.3.31 Register (REG)....174
6.3.32 RETURN....175
6.3.33 Recompile....176

6.3.34 Recompile without user comment....177
6.3.35 Search Forward, Search Backward (SF, SB)....181
6.3.36 STRING functions....182
6.3.37 WHILE/UNTIL loops....188
6.3.38 Cyclic execution of scripts: START_TIMER, STOP_TIMER....190

**7 Graphic and logic elements....193**

7.1 Line, dividing line, rectangle, circle and ellipse....193

7.2 Defining an array....196
7.2.1 Accessing the value of an array element....197
7.2.2 Example Access to an array element....198
7.2.3 Scanning the status of an array element....200

7.3 Table description (grid)....202
7.3.1 Defining a table (grid)....203
7.3.2 Defining columns....204
7.3.3 Focus control in the table (cell)....205

7.4 Custom widgets....206
7.4.1 Defining custom widgets....206
7.4.2 Structure of the custom widget library....206
7.4.3 Structure of the custom widget interface....207
7.4.4 Interaction between custom widget and dialog box - Automatic data exchange....209
7.4.5 Interaction between custom widget and dialog box - Manual data exchange....210
7.4.5.1 Reading and writing properties....210
7.4.5.2 Executing a method of the custom widget....212
7.4.5.3 Response to a custom widget signal....214

7.5 SIEsGraphCustomWidget....217
7.5.1 SIEsGraphCustomWidget....217
7.5.2 Notes regarding performance....219
7.5.3 Reading and writing properties....219
7.5.4 Properties....220
7.5.5 Functions....230
7.5.6 Signals....245

7.6 SIEsTouchButton....247
7.6.1 SIEsTouchButton....247
7.6.2 Reading and writing properties....248
7.6.3 Properties....249
7.6.4 Functions....265
7.6.5 Signals....266
7.6.6 Positioning and aligning picture and text....269
7.6.7 Language-dependent texts....271

**8 "Custom" operating area....273**

8.1 How to activate the "Custom" operating area....273

8.2 How to configure the "Custom" softkey....274

8.3 How to configure the "Custom" operating area....275

8.4 Programming example for the "Custom" area....276

**9 Dialog selection....281**
9.1 Dialog selection using PLC softkeys....281
9.2 Dialog selection using PLC hard keys....283
9.3 Dialog selection via NC....286
**10 Examples for cycle masks....287**
10.1 Examples for cycle masks....287
**11 Configuration in the side screen....289**
11.1 Display of a Run MyScreens configuration in a side screen page....290
11.2 Display of several Run MyScreens configurations in a side screen page....292
11.3 Adding Run MyScreens configurations to a side screen page....294
11.4 Adding Run MyScreens configurations to a side screen element....296
11.5 Notes on carrying out Run MyScreens configurations in the side screen....298
**12 Configuration in the Display Manager....299**
12.1 Displaying Run MyScreens configurations in the Display Manager....299
12.2 Integration in SISideScreenDialog....300
12.3 Integration into SIWidgetsApp....301
12.4 Notes on carrying out Run MyScreens configurations in the Display Manager....302
**A Reference lists....303**
A.1 Lists of start softkeys....303
A.1.1 List of start softkeys for turning....303
A.1.2 List of start softkeys for milling....304
A.2 List of predefined softkeys....307
A.3 List of access levels....308
A.4 List of colors....309
A.5 List of language codes used in file names....310
A.6 List of accessible system variables....311
A.7 Behavior when opening the dialog (attribute CB)....312
**B Tips and tricks....313**
B.1 General tips....313
B.2 Tips for debugging....315
B.3 Tips for the CHANGE method....316
B.4 Tips for DO LOOP loops....318
**C Animated elements....319**
C.1 Introduction....319
C.2 Modeling....321
C.2.1 Requirements....321

C.2.2 Rules for modeling....321
C.2.3 Importing graphics (models)....323
C.2.4 Modeling templates....325
C.3 XML commands....327
C.3.1 Overview....327
C.3.2 Structure of the scene description file....327
C.3.3 Mirroring and rotations....330
C.3.4 View type....330
C.4 Conversion to hmi file....332
C.5 Display in Create MyHMI /3GL....333
C.5.1 X3D Viewer....333
C.5.2 Class SIX3dViewerWidget....333
C.5.3 Public methods....333
C.5.4 Public slots....334
C.5.5 Libraries....335
C.5.6 Implementation example....335
C.5.7 Machine data....335
C.5.8 Notes about use....335
C.6 Display in Run MyScreens....336
**Glossary....337**
**Index....341**

# Introduction 1

## Overview

"Run MyScreens" can be used to create user interfaces that display functional expansions designed by the machine manufacturer or user, or to implement your own layout.

Cycle calls can also be generated with user interfaces that you have created. Dialogs can be created directly on the control system.

"Run MyScreens" is implemented with an Interpreter and configuration files that describe the user interfaces.

"Run MyScreens" is configured using ASCII files: These configuration files contain the description of the user interface. The syntax that must be applied in creating these files is described in the following chapters.

## Basic configuration

The "Run MyScreens" function enables machine manufacturers to configure their own dialogs. Even with the basic configuration, it is possible to configure five dialogs in the operator menu tree or for customer-specific cycle dialogs.



**Software option**

To expand the number of dialogs, you require one of the following software options:

- SINUMERIK 828D/840D sl, SINUMERIK Integrate Run MyScreens (6FC5800-0AP64-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyScreens + Run MyHMI (6FC5800-0AP65-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyHMI / 3GL (6FC5800-0AP60-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyHMI / WinCC (6FC5800-0AP61-0YB0)

## Supplementary conditions

The following conditions must be met:

- It is only possible to switch between dialogs within a single operating area.
- User variables may not have the same names as system or PLC variables (also see List Manual System Variables /PGAsl/).
- The dialogs activated by the PLC form a separate operating area (similar to measuring cycle screens).

## Tools

- UTF8-capable editor (e.g. the integrated editor of the user interface or Notepad)
- A graphics program is needed to create graphics/display images.

## File names and coding

**Note**

When using HMI Operate in the NCU, note that all file names are written in lower case letters (com, png, txt) on the CF card. This is required because of Linux.

File names are not case-sensitive on the PCU. However, it is recommended that you also use lower case letters here, in case of a possible subsequent transfer to Linux.

**Note**

When saving configuration and language files, ensure that the coding is set to UTF-8 for the editor you are using.

## Storage paths

Note the following convention when storing the configuration files, language files, etc.

| | | |
|---|---|---|
| *[System Siemens directory]* | | |
| | Linux: | /card/siemens/sinumerik/hmi |
| | Windows 7: | C:\ProgramData\Siemens\MotionControl\siemens |
| *[System oem directory]* | | |
| | Linux: | /card/oem/sinumerik/hmi |
| | Windows 7: | C:\ProgramData\Siemens\MotionControl\oem |
| *[System user directory]* | | |
| | Linux: | /card/user/sinumerik/hmi |
| | Windows 7: | C:\ProgramData\Siemens\MotionControl\user |
| *[System add_on directory]* | | |
| | Linux: | /card/addon/sinumerik/hmi |
| | Windows 7: | C:\ProgramData\Siemens\MotionControl\addon |
| | | |
| Storage locations | | |
| | "Run MyScreens" configuration: | *[System oem directory]*/proj |
| | Configuration file: | *[System oem directory]*/cfg |
| | Language files: | *[System oem directory]*/lng |

| | | |
|---|---|---|
| | Image files: | *[System oem directory]*/lco/ico*[Resolution]*<br>Example:<br>*[System oem directory]*/lco/ico640 |
| | Online Help: | *[System oem directory]*/hlp/*[Language]*<br>Example:<br>*[System oem directory]*/hlp/eng |

## Application

You can implement the following functions:

- Display dialogs containing the following elements:
  - Softkeys
  - Variables
  - Texts and Help texts
  - Graphics and Help displays
- Open dialogs by:
  - Pressing the (start) softkeys
  - Selection from the PLC/NC
- Restructure dialogs dynamically:
  - Edit and delete softkeys
  - Define and design variable fields
  - Insert, exchange and delete display texts (language-dependent or independent)
  - Insert, exchange and delete graphics
- Initiate operations in response to the following actions:
  - Displaying dialogs
  - Input values (variables)
  - Select a softkey
  - Exiting dialogs
  - Value change of a system or user variable
- Data exchange between dialogs:
- Variables:
  - Read (NC, PLC and user variables)
  - Write (NC, PLC and user variables)
  - Combine with mathematical, comparison or logic operators

- Execute functions:
  - Subprograms
  - File functions
  - PI services
- Apply protection levels according to user classes

# Getting Started

# 2

## 2.1 Introduction

Using the following example, you get to know the steps necessary to insert your own dialogs into the SINUMERIK Operate user interface using Run MySreens. You also learn how to create your own dialogs, insert context-sensitive help screens and help calls, define softkeys and how you can navigate between the dialogs.

# 2.2 Project example

## 2.2.1 Task description

In the project example you create the dialogs described in the following, including a context-sensitive help.

### Dialog 1

R parameters that can be written to (0 and 1) and geometry axis names are displayed in the first dialog (input fields). The corresponding help screens are linked in for the two R parameters. A context-sensitive help is linked in for the geometry axes. Further, the dialog includes examples for separating lines (horizontal and vertical), toggle fields, input/output fields with integrated unit selection and progress bars (with and without color change).

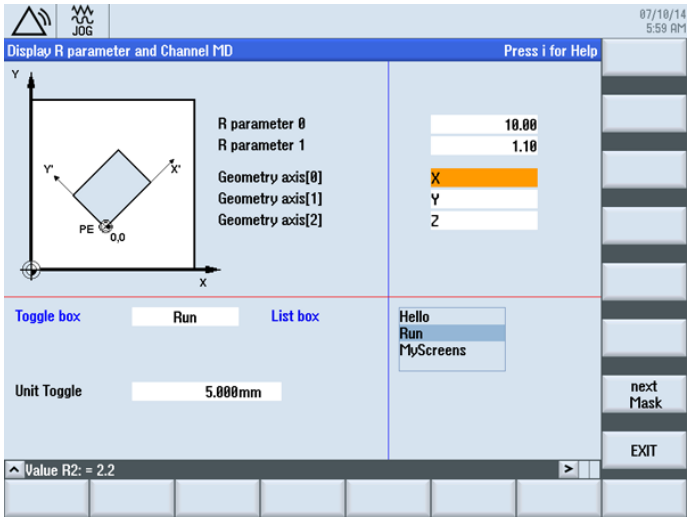



Figure 2-1 R parameters with help displays

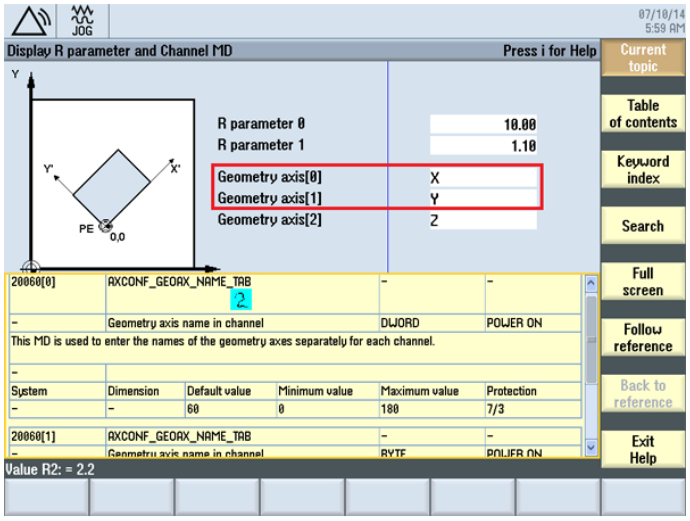



Figure 2-2 Geometry axes with context sensitive help

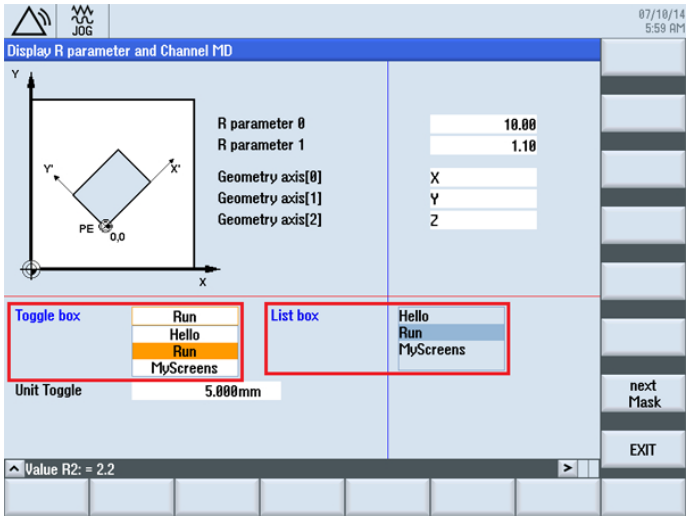



Figure 2-3 Toggle fields: List and box

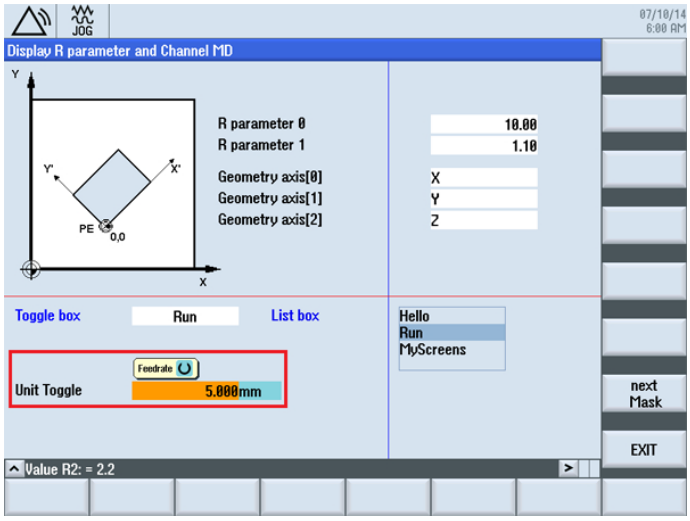



Figure 2-4 I/O field with integrated unit selection

## Dialog 2

MCS and WCS values are displayed in the second dialog. The dialog also contains examples of progress bars with and without color change.

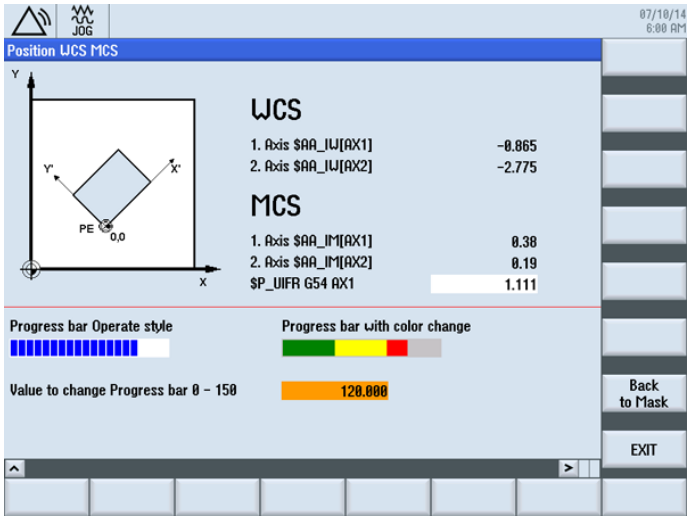



Figure 2-5 Progress bars with (right) and without (left) color change

## Navigation

The first dialog is called using the "START" softkey in the diagnostics operating area. The horizontal SK7 softkey is used.

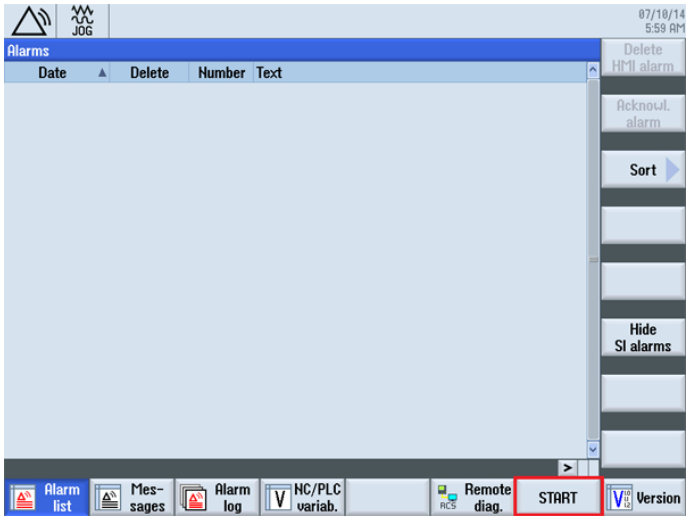



Figure 2-6 Start softkey „START“ in the machine operating area, AUTO mode

You can call the second dialog from the first dialog with the "Next Mask" softkey. You can return to the root screen of the operating area by pressing the "EXIT" softkey (see the screen above).

Using the "EXIT" softkey, you can also return to the root screen of the operating area from the second dialog (see the screen above). You can return to the first dialog via the "Back to Mask" softkey.

## Procedure

The necessary steps are described in the following chapters:

1. Creating the configuration file (COM file) (Page 18)
2. Saving the configuration file in the OEM directory (Page 21)
3. Creating the online help (Page 22)
4. Integrating the online help and saving the files to the OEM directory (Page 25)
5. Copying "easyscreen.ini" into the OEM directory (Page 26)
6. Registering the COM file in "easyscreen.ini" (Page 26)
7. Testing the project (Page 26)

## 2.2.2 Creating the configuration file (example)

### Content of the configuration file

Using a UTF8-capable editor, create the configuration file "diag.com" for the two dialogs.

```
; Start identifier of start softkey
//S(START)
   ; Start softkey only text
   HS7=("START")
   ; Start softkey with language-dependent text and png
   ;HS7=([$80792,"\\sk_ok.png"])

   ; Press method
   PRESS(HS7)
      ; LM or LS function
      LM("MASK1")
      ; LM, specifying a com file
      LM("MASK1","TEST.COM")
   END_PRESS
; End identifier of start softkey
//END

; Definition of dialog 1 with header and screen
//M(MASK1/"Display R parameter and Channel MD"/"mz961_01.png")
   ; Definition of the variables
   DEF VAR1 = (R2///,"R parameter 0"///"$R[0]"/200,50,150/400,50,100,)
   ; With help screen
   DEF VAR2 = (R2///,"R parameter 1"//"mz961_02.png"/"$R[1]"/200,70,150/400,70,100)
   ; With online help
   DEF ACHS_NAM1 = (S///"Press i for Help","Geometry axis[0]"////200,100,150/400,100,100//"sinumer-
ik_md_1.html","20060[0]")
   ; With online help
   DEF ACHS_NAM2 = (S///"Press i for Help","Geometry axis[1]"////200,120,150/400,120,100//"sinumer-
ik_md_1.html","20060[1]")
   DEF ACHS_NAM3 = (S///,"Geometry axis[2]"////200,140,150/400,140,100)
   ; Definition of toggle fields and unit toggle
   DEF VAR_TGL = (S/* "Hello", "Run", "MyScreens"/"Run"/,"Toggle box"////10,230,100/120,230,100/,,6)

   DEF VAR_TGB = (S/* "Hello", "Run", "MyScreens"/"Run"/,"List box"/
dt4///250,230,100/370,230,100,60/,,"#0602ee")
   ; BC, FC, BC_ST, FC_ST, BC_GT, FC_GT, BC_UT, FC_UT, SC1, SC2
   DEF VarEdit = (R///,"Unit Toggle",,,"Feedrate"///"$R[11]"/5,300,100/120,300,100///"VarTgl"),
VarTgl = (S/*0="mm",1="inch"/0//WR2////220,300,40)
```

```
; Softkey definition in the dialog
HS1=("")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")
VS2=("")
VS3=("")
VS4=("")
VS5=("")
VS6=("")
VS7=("Next Mask")
VS8=("EXIT")

; Definition LOAD block
LOAD
   ; Read value with RNP
   ACHS_NAM1 = RNP("$MC_AXCONF_GEOAX_NAME_TAB[0]")
   ACHS_NAM2 = RNP("$MC_AXCONF_GEOAX_NAME_TAB[1]")
   ACHS_NAM3 = RNP("$MC_AXCONF_GEOAX_NAME_TAB[2]")
   ; Output of a dialog line
   DLGL("Value R2: = " << RNP("$R[2]"))
   ; Write WNP to a value
   WNP("$R[3]",VAR0)

  ; Vertical dividing line
  V_SEPARATOR(360,1,6,1)
  ; Horizontal dividing line
  H_SEPARATOR(220,1,7,1)
END_LOAD

; Press method
PRESS(VS7)
   ; Load an additional dialog
   LM("MASK2")
; End identifier press method
END_PRESS

; Press method
PRESS(VS8)
```

```
      ; Exit the dialog
      EXIT
   ; End identifier press method
   END_PRESS
; End identifier dialog 1
//END

; Definition of dialog 2 with header and screen
//M(MASK2/"Position WCS MCS"/"mz961_01.png")
   ; Definition of the variables
   DEF TEXT1 = (I///,"WCS"/WR0,fs2///230,30,120/,,1)
   DEF VAR1 = (R3///,"1. Axis $AA_IW[AX1]"/WR1//"$AA_IW[AX1]"/230,70,150/400,70,100)
   DEF VAR2 = (R3///,"2. Axis $AA_IW[AX2]"/WR1//"$AA_IW[AX2]"/230,90,150/400,90,100)

   DEF TEXT2 = (I///,"MCS"/WR0,fs2///230,120,120/,,1)
   DEF VAR3 = (R2///,"1. Axis $AA_IM[AX1]"/WR1//"$AA_IM[AX1]"/230,160,150/400,160,100)
   DEF VAR4 = (R2///,"2. Axis $AA_IM[AX2]"/WR1//"$AA_IM[AX2]"/230,180,150/400,180,100)
   DEF VAR5 = (R3///,"$P_UIFR G54 AX1"///"$P_UIFR[1,AX1,TR]"/230,200,150/400,200,100)

   ; Definition of the progress bar
   DEF PROGGY0 = (R/0,150//,"Progress bar Operate style"/DT2,DO0//"$R[10]"/5,240,190/5,260,150/6,10)
   DEF PROGGY4 = (R/0,150,50,100/120/,"Progress bar with color change"/DT1,DO0//"$R[10]"/
260,240,190/260,260,150/3,4,,,9,7)

   ; Definition of the variables
   DEF PROGVAL = (R/0,150//,"Value to change Progress bar 0 - 150"///"$R[10]"/5,300,230/260,300,100)

  ; Softkey definition in the dialog
  HS1=("")
  HS2=("")
  HS3=("")
  HS4=("")
  HS5=("")
  HS6=("")
  HS7=("")
  HS8=("")
  VS1=("")
  VS2=("")
  VS3=("")
  VS4=("")
  VS5=("")
  VS6=("")
  VS7=("Back to Mask")
  VS8=("EXIT")
```

```
   ; Load method
   LOAD
      H_SEPARATOR(230,1,7,1)
   END_LOAD

   ; Change method
   CHANGE(VAR5)
      ; Function PI_START
      PI_START("/NC,201,_N_SETUFR")
   ; End identifier change method
   END_CHANGE

   ; Press method
   PRESS(RECALL)
      ; Return to the calling mask
      LM("MASK1")
   ; End identifier press method
   END_PRESS

   ; Press method
   PRESS(VS7)
      ; Return to the calling mask
      LM("MASK1")
   ; End identifier press method
   END_PRESS

   ; Press method
   PRESS(VS8)
      ; Exit the dialog to the standard application
      EXIT
   ; End identifier press method
   END_PRESS
; Dialog end identifier
//END
```

### 2.2.3 Saving the configuration file in the OEM directory (example)

#### Storage path

Save the configuration file "diag.com" under the following path:

*[System oem directory]*\proj

### 2.2.4 Creating the online help (example)

Create the HTML file "sinumerik_md_1.html". The example under "Content of the online help" shows the context-sensitive help call via **name="20060[0]"** and **name="20060[1]"**.

#### Notes

You can find notes on creating HTML files in Chapter Generating HTML files (Page 72).

Notes regarding the integration of online help are provided in Section Integrating the online help and saving the files to the OEM directory (example) (Page 25).

### Content of the online help

The example has no comments.

```
<html><head><meta http-equiv="Content-Type" content="text/html; charset="UTF-8"/><title></
title></head>
   <body>
      <table border="1" cellspacing="0" cellpadding="2" width="100%">
         <tr>
            <td><a name="20060[0]"><b>20060[0]</b></a></td>
            <td colspan="3">AXCONF_GEOAX_NAME_TAB</td>
            <center>
               <img src="../picture/bitmap3.png">
            </center>
            <td>-</td>
            <td>-</td>
         </tr>
         <tr>
            <td>-</td>
            <td colspan="3">Geometry axis name in channel</td>
            <td>DWORD</td>
            <td>POWER ON</td>
         </tr>
         <tr>
            <td colspan="6">This MD is used to enter the names of the geometry axes
separately for each channel.<br />
         <tr>
            <td>-</td>
            <td colspan="5"> </td>
         </tr>
         <tr>
          <td width="16*">System</td><td width="16*">Dimension</td><td width="16*">Default
value</td>
            <td width="16*">Minimum value</td><td width="16*">Maximum value</td>
            <td width="16*">Protection</td>
         </tr>
         <tr>
            <td>-</td>
            <td>-</td>
            <td>60</td>
            <td>0</td>
            <td>180</td>
            <td>7/3</td>
         </tr>
      </table>
      <p></p>
      <table border="1" cellspacing="0" cellpadding="2" width="100%">
         <tr>
            <td><a name="20060[1]"><b>20060[1]</b></a></td>
            <td colspan="3">AXCONF_GEOAX_NAME_TAB</td>
            <td>-</td>
            <td>-</td>
         </tr>
         <tr>
            <td>-</td>
            <td colspan="3">Geometry axis name in channel</td><td>BYTE</td>
            <td>POWER ON</td>
         </tr>
         <tr>
```

```
            <td colspan="6">This MD is used to enter the names of the geometry axes
separately for each channel.<br />
         <tr>
            <td>-</td>
            <td colspan="5"> </td>
         </tr>
         <tr>
            <td width="16*">System</td>
            <td width="16*">Dimension</td>
            <td width="16*">Default value</td>
            <td width="16*">Minimum value</td>
            <td width="16*">Maximum value</td>
            <td width="16*">Protection</td>
         </tr>
         <tr>
            <td>-</td>
            <td>-</td>
            <td>0</td>
            <td>0</td>
            <td>2</td>
            <td>7/3</td>
         </tr>
      </table>
      <p></p>
   </body>
</html>
```

## 2.2.5 Integrating the online help and saving the files to the OEM directory (example)

### Integrating the online help

The syntax to integrate the online help is analogous to SINUMERIK Operate.

DEF RFP=(R//1/,"RFP","RFP"///////"sinumerik_md_1.html","9006")

**Note**

As a result of LINUX, the HTML file must be written in lower case letters!

### Storage path

Save the HTML file "sinumerik_md_1.html" for the English help under the following path:

*[System oem directory]*hlp/eng

You must create a folder for additional languages (e.g. chs, deu, esp, fra, ita …).

A list of language codes is provided in the Appendix "Reference lists".

### Further information

Detailed information about creating OEM-specific online help files is provided in Chapter Configuring the online help (Page 71).

## 2.2.6 Copying "easyscreen.ini" into the OEM directory (example)

### Storage path

Copy the file "easyscreen.ini" from the directory

*[System SIEMENS directory]*cfg

to the directory

*[System oem directory]*cfg

## 2.2.7 Registering the COM file in "easyscreen.ini" (example)

### Adaptation of the "easyscreen.ini"

Make the following change in the "easyscreen.ini" in the OEM directory. You have thus registered the "diag.com" configuration file.

```
;########################
;# AREA Diagnosis       #
;########################
;<=============================================================>
;< OEM Softkey on first horizontal Main Menu                    >
;< SOFTKEY position="7"                                         >
;<=============================================================>
StartFile28 = area := AreaDiagnosis, dialog:= SlDgDialog, menu := DgGlobalHu, startfile := diag.com
```

## 2.2.8 Testing the project (example)

### Testing the dialog call

Switch to the diagnostics operating area. Click the "START" horizontal softkey.

If "Run MyScreens" detects errors when interpreting the configuration files, these errors will be written to the "easyscreen_log.txt" text file. Further information can be found in Chapter Troubleshooting (log book) (Page 36).

### Testing context sensitive help screens and online help

Test the context sensitive help screens and the online help. If errors occur, then check the paths where files have been saved.

# Fundamentals 3

## 3.1 Structure of configuration file

### Introduction

The description of new user interfaces is stored in configuration files. These files are automatically interpreted and the result displayed on the screen. Configuration files are not stored in the software supplied and must be set up by the user.

A UTF-8-capable editor (e.g. the integrated editor of the user interface or Notepad) is used to create the configuration files and language files. The description can also be explained using comments. A ";" is inserted as comment character before every explanation.

**Note**

When saving configuration and language files, ensure that the coding is set to UTF-8 for the editor you are using.

However, keywords - also in an UTF-8-coded COM file - may only comprise characters that are contained in the ASCII character set. Otherwise the interpretation and therefore the display of the screen forms/dialogs cannot be guaranteed.

### Configuration

Each HMI operating area has permanent start softkeys which can be used to access newly generated dialogs.

In the event of a "Load a screen form" (LM) or a "Load softkey menu" (LS) call in a configuration file, a new file name containing the object called can be specified. This makes it possible to structure the configuration, e.g. all functions in one operating level in a separate configuration file.

### Structure of the configuration file

A configuration file consists of the following elements:

1. Description of the start softkeys
2. Definition of the dialogs
3. Definition of the variables
4. Description of the methods
5. Definition of the softkey menus

**Note**

**Sequence**

The specified sequence in the configuration file must be maintained.

Example:

```
//S (START)                  ; Definition of the start softkey (optional)
....
//END
//M (.....)                  ; Definition of the dialog
DEF .....                    ; Definition of the variables
LOAD                         ; Description of the blocks
...
END_LOAD
UNLOAD
...
END_UNLOAD
...
//END
//S (...)                    ; Definition of a softkey menu
//END
```

## Storage location for configuration files

The configuration files are located in the *[System user directory]*/proj directory and also correspondingly stored in the *[System add_on directory]* and *[System oem directory]* directories.

## Converting texts from other HMI applications

Procedure to convert a text file with code page coding to text-coding UTF-8:

1. Open the text file on a PG/PC in a text editor.
2. When saving, set the UTF-8 coding.

The read-in mechanism via code page code is still supported.

# 3.2 Structure of the menu tree

## Menu tree principle

Several interlinked dialogs create a menu tree. A link exists if you can switch from one dialog to another. You can use the newly defined horizontal/vertical softkeys in this dialog to call the preceding or any other dialog.

A menu tree can be created behind each start softkey:

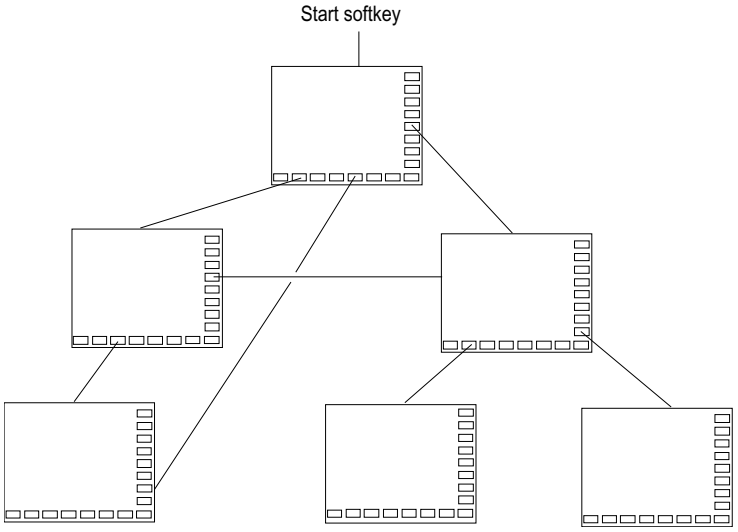



Figure 3-1 Menu tree

## Start softkeys

One or more softkeys (start softkeys), which are used to initiate your own operating sequences, are defined in the configuration files specified in "easyscreen.ini".

The loading of a dedicated dialog is associated with a softkey definition or another softkey menu. These are then used to perform the subsequent actions.

Pressing the start softkey loads the assigned dialog. This will also activate the softkeys associated with the dialog. Variables will be output to the standard positions unless specific positions have been configured.

## Reverting to the standard application

You can exit the newly created dialog and return to the standard operating area.

You can use the <RECALL> key to close new user interfaces if you have not configured this key for any other task.

**Note**

**Calling dialogs in the PLC user program**

Dialogs can be selected from the PLC as well as via softkeys: An interface signal is available in DB19.DBB10 for signal exchange between the PLC → HMI.

# 3.3 Definition and functions for start softkeys

## 3.3.1 Defining the start softkey

### Dialog-independent softkey

Start softkeys are dialog-independent softkeys which are not called from a dialog, but which have been configured **before** the first new dialog. In order to access the start dialog or a start softkey menu, the start softkey must be defined.

### Programming

The definition block for a start softkey is structured as follows:

```
//S(Start)                    ; Start identifier of start softkey
HS1=(...)                     ; Define the start softkey: Horizontal SK 1
PRESS(HS1)                    ; Method
  LM...                       ; LM or LS function
END_PRESS                     ; End of method
//END                         ; End identifier of start softkey
```

### Permissible positions for start softkeys

The following positions for "Run MyScreens" start softkeys are permissible in the operating areas:

| Operating area | Position |
|---|---|
| Machine | HSK6 |
| Parameter | HSK7 |
| Program | HSK6 and HSK15<br>Measuring cycles: HSK13 and HSK14 |
| Program manager | HSK2-8 and HSK12-16, if not assigned to drives. |
| Diagnostics | HSK7 |
| Startup | HSK7 |

Start softkeys are configured in special files. The names of these files are stated in the "easyscreen.ini" file. They usually have a name which is specific to an operating area (e.g. "startup.com" operating area for the startup area). The machine operating area represents an exception. There are several mode-specific files in the machine operating area ("ma_jog.com", "ma_auto.com").

The softkey menu with the start softkeys is called "Start". Existing configurations for start softkeys can still be used. The function whereby start softkeys are merged with the softkeys for the respective HMI application (operating area) in the start softkey menu is not supported.

This means that until the first dialog call is made - in other words, the time at which full functionality becomes available (e.g. execution of PRESS blocks) - menus or softkey menus can only be replaced by others in their entirety.

### Template for configuration of the start softkey

A detailed description of all permissible positions for start softkeys and their configuration is located in the "easyscreen.ini" file in the following directory:

*[System SIEMENS directory]*cfg

This file is used as a template for your own configuration of the start softkey.

### See also

Lists of start softkeys (Page 303)

## 3.3.2 Functions for start softkeys

### Functions for dialog-independent softkeys

Only certain functions can be initiated with start softkeys.

The following functions are permitted:

- The **LM function** can be used to load another dialog: **LM***("Identifier"[,"File"])*
- The **LS function** can be used to display another softkey menu: **LS***("Identifier"[, "File"][, Merge])*
- You can use the **"EXIT" function** to exit newly configured user interfaces and return to the standard application.
- You can use the **"EXITLS" function** to exit the current user interface and load a defined softkey menu.

### PRESS method

The softkey is defined within the definition block and the "LM" or "LS" function is assigned in the PRESS method.

If the start softkey definition is designated as a comment (semicolon (;) at beginning of line) or the configuration file removed, the start softkey will not function.

```
//S(Start)                        ; Start identifier
HS6=("1st screen form")           ; Horizontal SK 6 labeled "1st screen form"
PRESS(HS6)                        ; PRESS method for horizontal SK 6
  LM("Screen form1")              ; Load screen form1 function, where screen form 1
                                  must be defined within the same file.
```

```
END_PRESS                          ; End of PRESS method
HS7=("2nd screen form")            ; Horizontal SK 7 labeled "2nd screen form"
PRESS(HS7)                         ; PRESS method for horizontal SK 7
  LM("Screen form2")               ; Load screen form2 function, where screen form2
                                   must be defined within the same file.
END_PRESS                          ; End of PRESS method
//END                              ; End identifier of entry block
```

## Example

```
HS1 = ("new softkey menu")
HS2=("no function")
PRESS(HS1)
  LS("Menu1")                        ;Load new softkey bar
END_PRESS
PRESS (HS2)                          ; Empty PRESS method
END_PRESS
```

## Various configurations of the start softkeys

Various configurations of the start softkeys are merged. In this case, initially the name of the file to be interpreted is read-out of "easyscreen.ini". A search is made for *.com files in the following directories:

- *[System user directory]*/proj
- *[System oem directory]*/proj
- *[System add_on directory]*/proj
- *[System Siemens directory]*/proj

The configurations included for the start softkeys are now merged to form a configuration, i.e. the individual softkeys are compared. If there are two or more configurations for a softkey, the higher order configuration is always transferred into the merge version.

Softkey menus or dialogs that are possibly included are ignored. If a softkey has a command without file information, e.g. LM ("test"), as the required softkey menu or dialog is contained in the same file, then the corresponding file name is supplemented in the internal merge version so that in this case, no changes are required. The merge configuration contained is then subsequently displayed.

# 3.4 Troubleshooting (log book)

## Overview

If "Run MyScreens" detects errors when interpreting the configuration files, these errors will be written to the "easyscreen_log.txt" text file. Only errors of the currently selected dialog are entered. Error entries from previously selected dialogs are deleted.

The file indicates:

- The action during which an error occurred.
- The line and column number of the first faulty character.
- The entire faulty line of the configuration file.
- The entries made by the DEBUG function.

**Note**

Each entry has a prefix with the current time stamp in square brackets. This can be helpful, for example, when analyzing time-critical configurations.

## Saving "easyscreen-log.txt"

The file "easyscreen_log.txt" is saved in the following directory:

*[System user directory]*\log

## Syntax

The system does not start to interpret the syntax until the start softkey has been defined and a dialog with start and end identifiers as well as a definition line has been configured.

```
//S(Start)
HS6=("1st screen")
PRESS(HS6)
  LM("Maske1")
END_PRESS
//END

//M(Maske1)
  DEF Var1=(R)
  DEF VAR2 = (R)
  LOAD
  VAR1 = VAR2 + 1             ; Error message in log book, as VAR2 has no value
...
//END
```

```
; The right entry would be,
for example:
//M(Maske1)
   DEF Var1=(R)
   DEF VAR2 = (R)

   LOAD
      VAR2 = 7
      VAR1 = VAR2 + 1 ;
...
```

# 3.5 Notes on "easyscreen.ini"

As of SINUMERIK Operate V4.7, "easycreen.ini" has been extended by the entries described in this chapter. The "easyscreen.ini" can be found in the *[System Siemens directory]*\cfg directory.

## Help screen start position

**Entry in "easyscreen.ini":**

```
[GENERAL]
HlpPicFixPos=true
```

**Note:**

The start position of help displays is positioned at the configured pixel position independent of the resolution (default=true).

## Elongation behavior, default line height and line spacing

**Entry in easyscreen.ini:**

```
[GENERAL]
SymmetricalAspectRatio=false
DefaultLineHeight=18
DefaultLineSpacing=3
```

**Notes:**

- The "SymmetricalAspectRatio" entry determines whether the same aspect ratio is used in the X and Y direction when adapting a configuration to a specific screen resolution.
  - "false" (default): Fields and graphics are shortened in widescreen resolution in the Y direction (asymmetrical elongation in relation to 640x480). For example, a square configured in 640x480 is shortened vertically for a widescreen panel and becomes a rectangle.
  - "true": The same aspect ratio is used in the X and Y direction and fields and graphics retain their originally configured proportions in relation to 640x480. For example, a square configured in 640x480 remains a square in a widescreen panel.
- The entries "DefaultLineHeight" and "DefaultLineSpacing" can be used to specify the default line height (default: 18 pixels) and the default line spacing (default: 3 pixels) in relation to 640x480. They only take effect when no Y position or height is specified in the configuration for the position of the short text or the input/output field.

## Cycles in the Machine main screen

- Entry to machine operating mode JOG via HS6 with the possibility to generate a cycle call into the Machine main screen with section [JOBSHOPINTEGRATION]:
  ```
  [JOBSHOPINTEGRATION]
  Integration = true
  ```
  or via the starting block in the configuration
  ```
  LM("screen form1",,1)
  PRESS(VS8)
     GC("MOVE_RIDE") ; cycle call is being generated
     EXIT
  END_PRESS
  ```
- Retention of the operate menus with section [Integration]:
  If the dialog only includes vertical softkeys, the horizontal standard softkey bar incl. the start softkey is retained when the dialog is displayed. If the dialog contains horizontal and vertical softkeys, the horizontal bar from the dialog replaces the horizontal bar with the start softkey.
  ```
  [Integration]
  OperateMenusEnabled = true
  ```

## Resolution-dependent screen position: Form panels

**Entry in "easyscreen.ini":**

```
[640x480]
MyPanel = x:=0, y:=220, width:=340, height:=174
[800x480]
MyPanel = x:=0, y:=220, width:=420, height:=174
...
```

**Notes:**

The screen position can be defined as follows:

- The screen position can be specified in "Pixels in relation to 640x480".
  Example:
  `//M(MyMask/"MyCaption"/"\\myhelp.png"/`**0,219,335,174**`)`
- The screen position can be coupled to the resolution-dependent position of a form panel from an Operate standard screen layout, e.g. "Form panel4" ("auxiliary functions") from the screen layout "slmastandardscreenlayout.SlMaStandardScreenLayout" of the "Machine" area
  `//M(MyMask/"MyCaption"/"\`
  `\myhelp.png"/`**"slmastandardscreenlayout.SlMaStandardScreenLayout.FormPanel4"**`)`

  It is therefore easy to superimpose standard forms from Operate or position Easyscreen screens precisely at their position.
  Example:
  `//M(Mask/"Mask"//`**"slstandardscreenlayout.SlStandardScreenLayout.LowerForm"**`)`
  Any other screen layout can also be used.
- The screen position can be coupled to a self-defined, resolution-dependent definition of form panels in the "easyscreen.ini" file.
  Example:
  `//M(MyMask/"MyCaption"/"\\myhelp.png"/`**"MyPanel"**`)`

# 3.6 Notes for personnel changing over to "Run MyScreens"

**Note**

When using HMI Operate in the NCU, note that all file names are saved in lower case letters on the CF card (com, png, txt).

**Note**

Make sure that the coding is set to UTF-8 in the editor that you are using when saving the configuration and language files.

## Image files

Always save image files in the PNG format "*.png".

The data must be saved, e.g. for OEM modifications, under

*[System oem directory]*/ico/[Resolution]

Further information can be found in Chapter Using display images/graphics (Page 61).

## Adapting the configuration file

Check your configuration files regarding the following points:

- Compare the start softkeys with the currently permissible softkeys and adapt them, if required.
- Rename the integrated image files corresponding to the "Image files" point above.

The data is stored, e.g. for OEM modifications, in the following directory:

*[System oem directory]*/proj

*[System user directory]*/proj

*[System add_on directory]*/proj

## Adapting the help files

All help files must be saved in UTF-8. Check your existing files, and resave them accordingly with a suitable editor.

The HTML files are saved in the following directory, e.g. for English:

*[System oem directory]*/hlp/eng

*[System user directory]*/hlp/eng

*[System add_on directory]*/hlp/eng

You must create the directories for additional languages corresponding to the language codes.

## Checking the "Run MyScreens" license

Check whether the number of dialogs that you have entered exceeds the basic scope of a maximum of five dialogs.

To expand the number of dialogs, you require one of the following software options:

- SINUMERIK 828D/840D sl, SINUMERIK Integrate Run MyScreens (6FC5800-0AP64-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyScreens + Run MyHMI (6FC5800-0AP65-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyHMI / 3GL (6FC5800-0AP60-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyHMI / WinCC (6FC5800-0AP61-0YB0)

# 3.7 Extended configuration syntax

As of SINUMERIK Operate V4.7, a simplified syntax is available for the definition of screens, variables, softkeys and for table columns. Reading and servicing are improved with this alternative syntax. The properties and attributes can be specified in any order, empty entries are omitted. Instead of with round brackets "**(**" and "**)**" as in the previous syntax, the listing of the properties and attributes is now enclosed in curly brackets "**{**" and "**}**".

The properties and attributes are specified as follows:
`{<Name> = <Value>, <Name> = <Value>, …}`

The previous syntax remains compatible.

## Extended syntax for the definition of screens

//M {<Mask name> [,**HD**=<Title>] [,**HLP**=<Graphic>] [,**X**=<X position>] [,**Y**=<Y position>] [,**W**=<Width>] [,**H**=<Height>] [,**VAR**=<> System or user variable] [,**HLP_X**=<X position help screen] [,**HLP_Y**=<Y position help screen>] [,**CM**=<Column alignment>] [,**CB**= Response when opening the dialog] [,**XG**=<Interpret help screen as X3d graphic>] [,**PANEL**=<Name of the linked FormPanel>][,**MC**=<Mask background color:>][,**HD_AL**=<Alignment of the mask header>] [,**LANGFILELIST**=<List of the mask-specific language files>]}

**Example:**
```
//M{VariantTest, HD="My Mask"}
```

## Extended syntax for the definition of variables

DEF <Variable name> = {[**TYP**Type] [,**MIN**=<Minimum value>] [,**MAX**=<Maximum value>] [,**TGL**=<Toggle values>] [,**VAL**=<Preassignment>] [,**LT**=<Long text>] [,**ST**=<Short text>] [,**GT**=<Graphic text>] [,**UT**=<Unit text>] [,**TT**=<ToolTip text>] [,**TG**=<Toggle option>] [,**WR**=<Input mode>] [,**AC**=<Access level>] [,**AL**=<Text alignment>] [,**FS**=<Font size>] [,**LI**=<Limit value handling>] [,**UR**=<Refresh rate>] [,**CB**=<Text align when opening the dialog>] [,**HLP**=<Help screen>] [,**VAR**=<System or user variable>] >] [,**TXT_X**=<X position short text>] [,**TXT_Y**=<Y position short text>] [,**TXT_W**=<Width short text>] [,**TXT_H**=<Height short text>] [,**X**=<X position input/output field>] [,**Y**=<Y position input/output field>] [,**W**=<Width input output field>] [,**H**=<Height input/output field>] [,**UT_DX**=<Distance between input field and unit field>] [,**UT_W**=<Width unit field>] [,**BC**=<Background color input/output field>] [,**FC**=<Foreground color input/output field>] [,**BC_ST**=<Background color short text>] [,**FC_ST**=<Foreground color short text>] [,**BC_GT**=<Background color graphic text>] [,**FC_GT**=<Foreground color graphic text>] [,**BC_UT**=<Background color unit text>] [,**FC_UT**=<Foreground color unit text>] [,**SC1**=<Signal color 1 for progress bar>] [,**SC2**=<Signal color 2 for progress bar>] [,**SVAL1**=<Threshold value 1 for progress bar>] [,**SVAL2**=<Threshold value 2 for progress bar>] [,**DT**=<Display type>] [,**DO**=<Display alignment>] [,**OHLP**=<Online help>] [,**LINK_TGL**=<Name of the linked toggle v]}

**Examples:**

```
DEF MyVar5={TYP="R2", ST="MyVar5", VAL=123.4567, OHLP="myhelp.html", MIN=100.1,
MAX=200.9}
DEF MyVar2={TYP="I", TGL="*1,2,3", VAL=1}
DEF MyVar3={TYP="R2", TGL="*0=""Off"", 1=$80000", VAL=1}
```

```
DEF MyVar4={TYP="R2", TGL="*MyArray",VAL=1}
DEF MyVar1={TYP="R2", TGL="%grid99", X = 0, W=300, H=200}
DEF MyVar6={TYP="R2", TGL="+$80000", VAR="$R[10]", ST="Text offset"}
```

## Extended syntax for the definition of softkeys

SK = {[**ST**=<Label>] [,**AC**=<Access level>] [,**SE**=<Status>]}

**Examples:**

```
HS1={ST="""MySk""", AC=6, SE=1}
HS3={ST="SOFTKEY_CANCEL"}
HS5={ST="[$81251,""\\sk_ok.png""]"}
HS8={ST="[""Test"",""\\sk_ok.png""]"}
```

## Extended syntax for the definition of table columns

{[**TYP**=<Type>] [,**MIN**=<Minimum value>] [,**MAX**=<Maximum value>] [,**LT**=<Long text>] [,**ST**=<Short text>] [,**WR**=<Input mode>] [,**AC**=<Access level>] [,**AL**=<Text alignment>] [,**FS**=<Font size>] [,**LI**=<Limit value handling>] [,**UR**=<Refresh rate>] [,**HLP**=<Help screen>] [,**VAR**=<System or user variable>] >] [,**W**=<Column width>] [,**OF1**=<Offset1>] [,**OF2**=<Offset2>] [,**OF3**=<Offset3>]}

**Example:**

```
DEF MyGridVar={TYP="R", TGL="%MyGrid1", X=10, W=550, H=100}
//G(MyGrid1/0/5)
   {TYP="I", ST="Index", WR=1, VAR="1", W=80, OF1=1}
   {TYP="S", LT="LongText2", ST="Text", WR=1, VAR="$80000", AL=2, W=330, OF1=1}
   {TYP="R3", LT="LongText1", ST="R9,R11,R13,R15", WR=2, VAR="$R[1]", W=110, OF1=2}
//END
```

## 3.8 SmartOperation and MultiTouch operation

You have the following setting options for specific adjustment to SmartOperation and for MultiTouch operation:

- Automatic adjustment of the field height and field width to the so-called minimum usable field size and the line spacing (mask attribute MA).
- Optimized and pixel-precise stretching of the fields for higher resolutions (mask attribute PA).
- Setting of the field height and line spacing proportional to the font (mask attribute FA).
- Allows free scrolling so that the input field is not covered by the virtual keyboard (mask attribute KM).

Further details can be found in Chapter Defining dialog properties (Page 49), Section "Programming".

# Dialogs

# 4

## 4.1 Structure and elements of a dialog

### 4.1.1 Defining a dialog

#### Definition

A dialog is part of a user interface consisting of a display line, dialog elements and/or graphics, an output line for messages and 8 horizontal and 8 vertical softkeys.

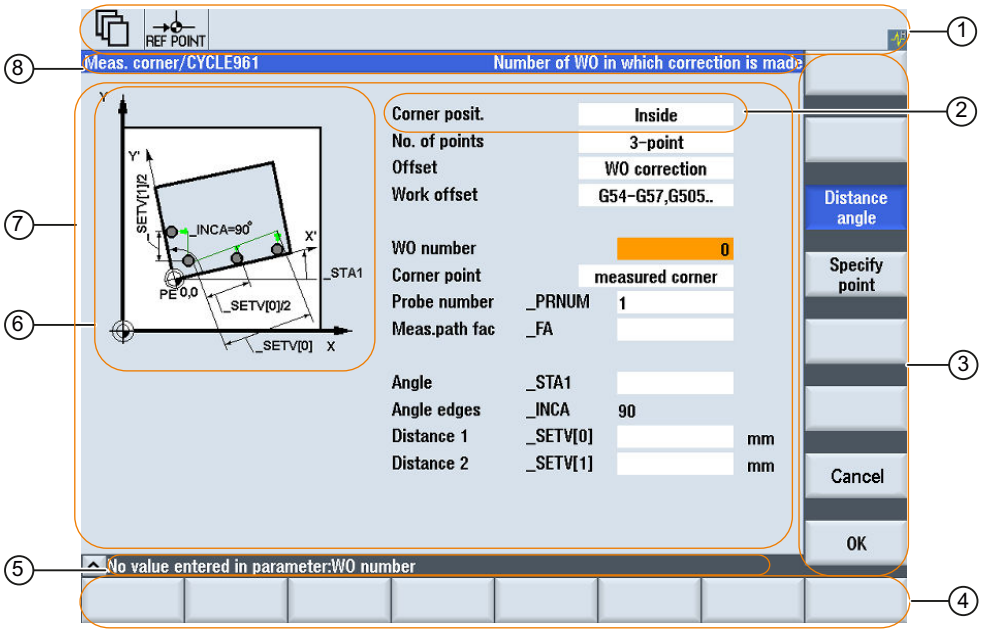
Dialog elements are:

- Variables
  - Limits/toggle field
  - Default setting of variables
- Help display
- Texts
- Attributes
- System or user variable
- Position of short text
- Position of input/output field
- Colors

Dialog properties:

- Header
- Graphic
- Dimension
- System or user variable

- Graphic position
- Attributes




① Machine status display ("header")
② Dialog element
③ 8 vertical softkeys
④ 8 horizontal softkeys
⑤ Output of diagnostic messages
⑥ Graphic
⑦ Dialog
⑧ Header line of the dialog with header and long text

Figure 4-1 Structure of the dialog

## Overview

The definition of a dialog (definition block) is basically structured as follows:

| Definition block | Comment | Chapter reference |
|---|---|---|
| `//M...` | ;Dialog start identifier | |
| `DEF Var1=...`<br>`...` | ;Variables | Variables (Page 81) |
| `HS1=(...)`<br>`...` | ;Softkeys | Defining softkey menus (Page 62) |
| `PRESS(HS1)`<br>`  LM...`<br>`END_PRESS` | ;Method start identifier<br>;Actions<br>;Method end identifier | Methods (Page 126) |
| `//END` | ;Dialog end identifier | |

Within the dialog definition block, various variables that appear as dialog elements in the dialog, as well as horizontal and vertical softkeys, are defined first. Different types of actions are then configured in methods.

**Note**

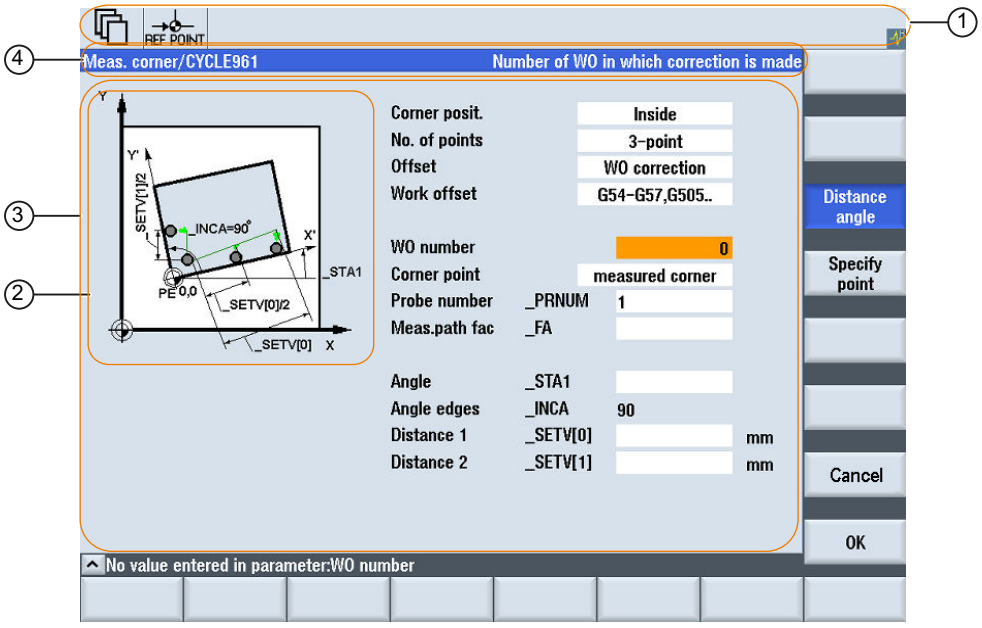
In operating area "Machine", for a dialog change, the following sequence should be observed:

Under the precondition that the dimensions of the following mask are smaller than the previous mask – or the position of the following mask is different than the previous mask, dialog change only functions if the following mask is exited before returning to the first mask, and only then is the first mask reloaded.

## 4.1.2 Defining dialog properties

### Description

The properties of the dialog are defined in the start identifier line of the dialog.




① Machine status display ("header")
② Graphic
③ Dialog
④ Header line of the dialog with header and long text

Figure 4-2 Dialog properties

## Programming

| | |
|---|---|
| Syntax: | **//M**(Identifier/[Title]/[Graphic]/[Dimension]/[System or user variable]/[Position graphic]/[Attribute]/List of the mask-specific language files)<br>See also Chapter Extended configuration syntax (Page 43). |
| Description: | Defines a dialog |

| | | | | |
|---|---|---|---|---|
| Parameter: | Identifier | | | Name of the dialog |
| | Header | | | Dialog header as text or call for text (e.g. $85011) from a language-specific text file. |
| | Graphic | | | Graphics file with path in double quotation marks. |
| | Dimension | | | Position and size of the dialog in pixels (distance from left-hand side, distance from top, width, height), in relation to the upper left-hand corner of the screen. The entries are separated by a comma. |
| | System or user variable | | | System or user variable to which the current cursor position is assigned. The NC or PLC can be provided with the cursor position via the system or user variable. The first variable has index 1. The order corresponds to the configuration order of the variables. |
| | Graphic position | | | Position of the graphic in pixels (distance from left-hand side, distance from top), in relation to the upper left-hand corner of the dialog. The entries are separated by a comma. |
| | Attributes | | | The specifications of the attributes are separated by a comma.<br>Possible attributes are: |
| | | CM | | Column mode: Column alignment |
| | | | CM0 | Default setting: The column distribution is carried out separately for each line. |
| | | | CM1 | The column distribution of the line with the most columns applies to all lines. |
| | | CB | | CHANGE block: Response when dialog is opened: cb attributes specified for a variable in a variables definition take priority over the default setting in the dialog definition.<br>See also Behavior when opening the dialog (attribute CB) (Page 312). |
| | | | CB0 | Default setting: All CHANGE blocks associated with the dialog are processed when it is opened. |
| | | | CB1 | CHANGE blocks are then only processed if the relevant value changes. |
| | | XG | | Integration of an X3D animation as help display (only in the machining step programming).<br>Example:<br>`//M(Meas/$85605/"myx3dhelpfile.hmi,,Z_Animation,,G17"///30,10/XG1)` |
| | | | XG0 | Default setting = 0 |
| | | | XG1 | The mask attribute XG must already be set to 1 in the mask definition, a change during runtime is not possible.<br>Note:<br>The specification in the mask property HLP must also be set accordingly. |
| | | AL | | The alignment of the mask header can be influenced with the mask attribute AL.<br>Example:<br>Central setting of the "Set password" window header:<br>`//M(MY_PWD_SET/”Set password”//"EasyPwdModalLayout"//AL2)` |
| | | | AL0 | Left-justified, default |
| | | | AL1 | Right-justified |
| | | | AL2 | Centered |

| | | |
|---|---|---|
| KM | | Freely scrolling in the mask (SlGfwScrollArea), if the virtual keyboard is to be displayed for MultiTouch, can be influenced as follows:<br>• In the mask definition line with mask attribute "KM" (KeyboardMode) Example:<br>`//M(MyMTMask/"MultiTouch Mask"/////KM1)`<br>• As global setting for all Run MyScreens masks in the "easyscreen.ini" Example:<br>`[GENERAL]`<br>`DefaultVirtualKeyboardMode=1`<br><br>Note:<br>If, when configuring the masks, the mask attribute KM is explicitly set, then this has a higher priority than the global setting in "easyscreen.ini".<br><br>(See also Chapter SmartOperation and MultiTouch operation (Page 45)) |
| | KM1 | Free scrolling active (default) |
| | KM0 | Free scrolling inactive |
| PG | | Alignment and appearance of the window title in conjunction with PNG images (only effective in the editor!)<br>Example:<br>`//M(MyPg1SampleMask/"My PG1 Sample Mask"/////PG1)` |
| | PG0 | (Default) |
| | PG1 | Alignment and appearance of the window title in conjunction with PNG images<br>Path specification (on the far left), mask name (on the right)<br>The display of the long text is no longer possible in this mode. Alternatively, you can work with tooltips. |
| MA | | Automatic adjustment of the field height for MultiTouch operation (Mutli-Touch Adjustment)<br>If required, the adjusment for a MultiTouch operator panel includes the necessary enlargement to the so-called minimum usable size (width, height) of the fields (exceptions: progress bar, animated GIF, CustomWidget) and the line spacing.<br>The MultiTouch adjustment can be made<br>• globally for all Run MyScreens mask in "easyscreen.ini", e.g.<br>`[GENERAL]`<br>`DefaultMultiTouchAdjustmentLevel=1`<br>• or mask-specifically in the mask definition line with the mask property "MA" in order to overwrite the global setting in "easyscreen.ini", e.g.<br>`//M(MyMTMask/"MultiTouch Mask"/////MA1)`<br>If the mask attribute MA is set explicitly in the mask configuration, then this has a higher priority than the global setting in "easyscreen.ini".<br>(See also Chapter SmartOperation and MultiTouch operation (Page 45)) |
| | MA0 | There is no adjustment |
| | MA1 | Only those fields are adjusted for which no spacing from the top and/or field height/width has been configured – i.e. the fields that use default positions and therefore for which Run MyScreens calculates the spacing from the top and/or field height/width. (Default) |

| | | |
|---|---|---|
| | MA2 | All fields are adjusted irrespective of the configured field sizes. |
| PA | | Optimized and pixel-precise stretching of the fields for higher resolutions (Pixel Fine Adjustment)<br>Previously all field positions were calculated in relation to 640x480 and then zoomed with the appropriate horizontal and vertical stretch factors. This process has the disadvantage that rounding errors can occur with "unfavorable stretch factors". For example, the field height or line spacing can "jump" by a pixel from line-to-line. The "Pixel Fine Adjustment" mode can be used to prevent this as the field positions are determined directly in the current resolution.<br>`//M(MyMask/"My Mask"/////PA1)`<br>This setting can also be made globally for all Run MyScreens masks in "easyscreen.ini", e.g.<br>`[GENERAL]`<br>`DefaultPixelFineAdjustment=1`<br>If the mask attribute PA is set explicitly in the mask configuration, it has a higher priority than the global setting in "easyscreen.ini".<br>If a mask is displayed with Font Adjustment = FA1, then the Pixel Fine Adjustment mode is also automatically active for this mask.<br>(See also Chapter SmartOperation and MultiTouch operation (Page 45)) |
| | PA0 | Stretching as previously, i.e. the positions are calculated in relation to 640x480 and then stretched.<br>(For compatibility reasons: Default) |
| | PA1 | Optimized and pixel-precise stretching of the fields for higher resolutions |
| FA | | Field height and line spacing proportional to the font (Font Adjustment)<br>This setting can also be made globally for all Run MyScreens masks in "easyscreen.ini", e.g.<br>`[GENERAL]`<br>`DefaultFontAdjustment=1`<br>If the mask attribute FA is set explicitly in the mask configuration, then this has a higher priority than the global setting in "easyscreen.ini".<br>The settings for the DefaultLineHeight and DefaultLineSpacing have no significance when Font Adjustment mode is active.<br>(See also Chapter SmartOperation and MultiTouch operation (Page 45)) |
| | FA0 | Field height and line spacing are stretched as previously. (For compatibility reasons: Default) |
| | FA1 | Field height and line spacing are set proportional to the font (automatically sets mask attribute PA=1) |
| | FA2 | Same as FA1, but the X coordinate and the field width are also stretched proportionally to the font<br>(Automatically sets mask attribute PA=1)<br>(Only possible in conjunction with attribute XG=1!) |
| NT | | Navigation Type |
| | NT0 | NT0 = standard mode<br>The navigation is analogous to the masks of the operating area in which Run MyScreens mask is integrated. This means in the cycle masks in the editor, navigation is line-oriented (see NT2), in all other "normal" masks, it is a geometrical (see NT1). |

| | | | | |
|---|---|---|---|---|
| | | | NT1 | Geometrical navigation (top, bottom, left, right), until the borders of the mask are reached (standard in a normal Run MyScreens environment, e.g. area "Custom") |
| | | | NT2 | Line-oriented, i.e. within the line to the right until the end of the line is reached (standard, in the cycle masks of the editor) |
| | | NR | | Navigation direction when pressing the Enter key (Return Navigate Direction) |
| | | | NR0 | NR0 = off (default), i.e. when pressing the Enter key, navigation is realized within the mask in a tabular sequence (standard in the normal Run MyScreens environment) |
| | | | NR1 | When pressing the Enter key, navigation is the same as when pressing the up arrow key |
| | | | NR2 | As for NR1, however, downward |
| | | | NR3 | As for NR1, however, to the left |
| | | | NR4 | As for NR1, however, to the right |
| | | TA | | Tooltip alignment |
| | | | TA0 | Automatic (default) |
| | | | TA1 | Left |
| | | | TA2 | Right |
| | | | TA3 | Top |
| | | | TA4 | Bottom |
| | | | TA5 | Top left |
| | | | TA6 | Bottom left |
| | | | TA7 | Top right |
| | | | TA8 | Bottom right |
| | MC | | | Mask background color (mask color)<br>Example:<br>Set mask background color blue (= 6)<br>`//M(MyMask/"MyMask"//////6)`<br>or<br>`PRESS(HS1)`<br>`  MC=6`<br>`END_PRESS` |
| | List of the mask-specific language files | | | separated by a comma |

## Accessing the dialog properties

Read and write access to the following dialog properties is permitted within methods (e.g. PRESS block).

- HD = Header
- HLP = Help display
- VAR = System or user variable
- MC = Mask background color
- CM = Column alignment (read-only)

- CB = Behavior when opening (read-only)
- XG = Integration of X3d (read-only)
- AL = Mask header alignment (read-only)

## Example

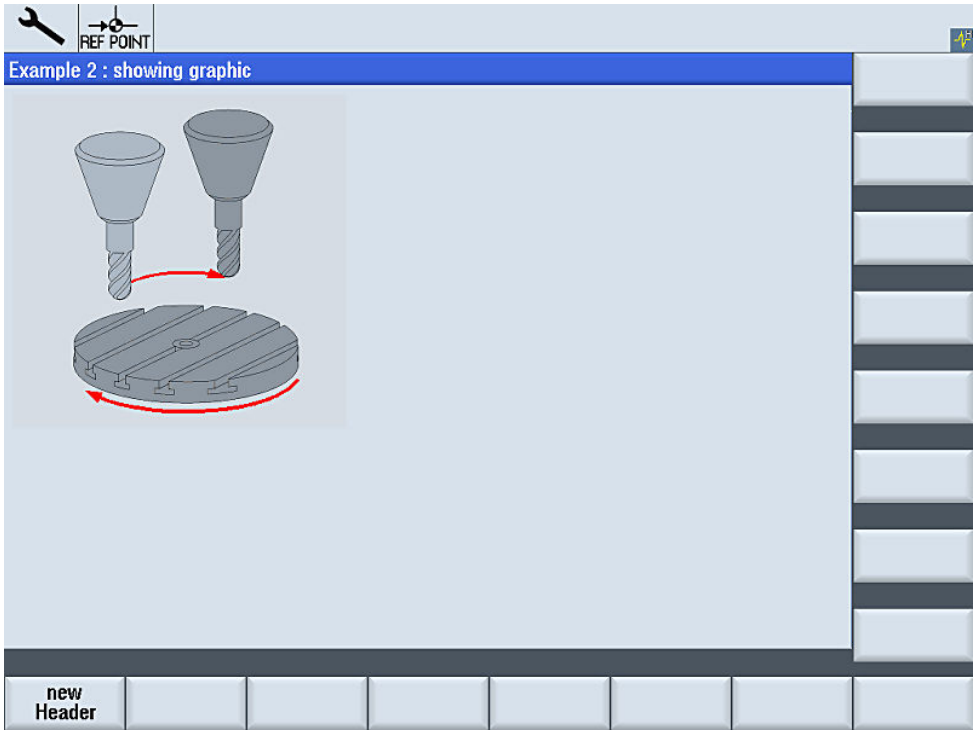



Figure 4-3 "Example 2: showing graphic"

```
//S(Start)
HS7=("Example", se1, ac7)

PRESS(HS7)
  LM("Mask2")
END_PRESS

//END
//M(Mask2/"Example 2 : showing graphic"/"example.png")
HS1=("new%nHeader")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
```

```
HS8=("")
VS1=("")
VS2=("")
VS3=("")
VS4=("")
VS5=("")
VS6=("")
VS7=("")
VS8=("")

PRESS(HS1)
  Hd= "new Header"
END_PRESS
...
//END
```

### See also

Programming example for the "Custom" area (Page 276)

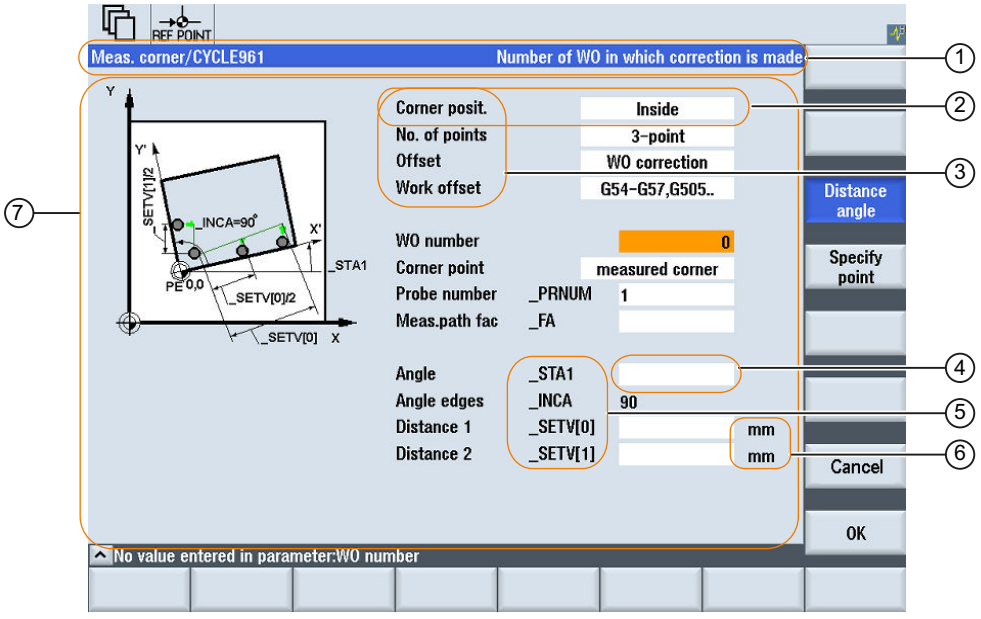
## 4.1.3 Defining dialog elements

### Dialog element

The term "dialog element" refers to the visible part of a variable, i.e. short text, graphics text, input/output field, unit text and tooltip. Dialog elements fill lines in the main body of the dialog. One or more dialog elements can be defined for each line.

## Variable properties

All variables are valid in the active dialog only. Properties are assigned to a variable when it is defined. The values of dialog properties can be accessed within methods (e.g. a PRESS method).




① Header line of the dialog with header and long text
② Dialog element
③ Short text
④ Input/output field
⑤ Graphic text
⑥ Text for units
⑦ Main body of the dialog

Figure 4-4 Elements of a dialog

## Programming - Overview

The single parameters to be separated by commas are enclosed in round brackets:

| DEF *Identifier* = | **Identifier = Name of variable** |
|---|---|
| | Variable type |
| | /[Limit values or toggle field] |
| | /[Default] |
| | /[Texts (Long text, Short text\|Image, Graphic text, Units text)] |
| | /[Attributes] |
| | /[Help display] |
| | /[System or user variable] |
| | /[Position of short text] |
| | /[Position of I/O field(Left, Top, Width, Height)] |

| | |
|---|---|
| | /[Colors] |
| | /[online help] |

### See also

Variable parameters (Page 92)

## 4.1.4 Defining dialogs with multiple columns

### Overview

Multiple variables can also be represented in a dialog on one line. In this case, the variables are all defined in the configuration file on a single definition line.

```
DEF VAR11 = (S///"Var11"), VAR12 = (I///"Var12")
```

To make individual variables in the configuration file more legible, the definition lines can be wrapped after every variables definition and following comma.

The key word "DEF" always indicates the beginning of a new line:

```
DEF Tnr1=(I//1/"","T ",""/wr1///,,10/20,,50),
  TOP1=(I///,"Type="/WR2//"$TC_DP1[1,1]"/80,,30/120,,50),
  TOP2=(R3///,"L1="/WR2//"$TC_DP3[1,1]"/170,,30/210,,70),
   TOP3=(R3///,"L2="/WR2//"$TC_DP4[1,1]"/280,,30/320,,70),
   TOP4=(R3///,"L3="/WR2//"$TC_DP5[1,1]"/390,,30/420,,70)
DEF Tnr2=(I//2/"","T ",""/wr1///,,10/20,,50),
   TOP21=(I///,"Typ="/WR2//"$TC_DP1[2,1]"/80,,30/120,,50),
   TOP22=(R3///,"L1="/WR2//"$TC_DP3[2,1]"/170,,30/210,,70),
   TOP23=(R3///,"L2="/WR2//"$TC_DP4[2,1]"/280,,30/320,,70),
   TOP24=(R3///,"L3="/WR2//"$TC_DP5[2,1]"/390,,30/420,,70)
```

**Note**

When configuring multi-column dialogs, please observe that a large number of columns might slow down the system!

## 4.1.5 Password dialogs

### Using the standard password dialogs

You can integrate the following predefined password dialogs in the screen configuration:

- Set password

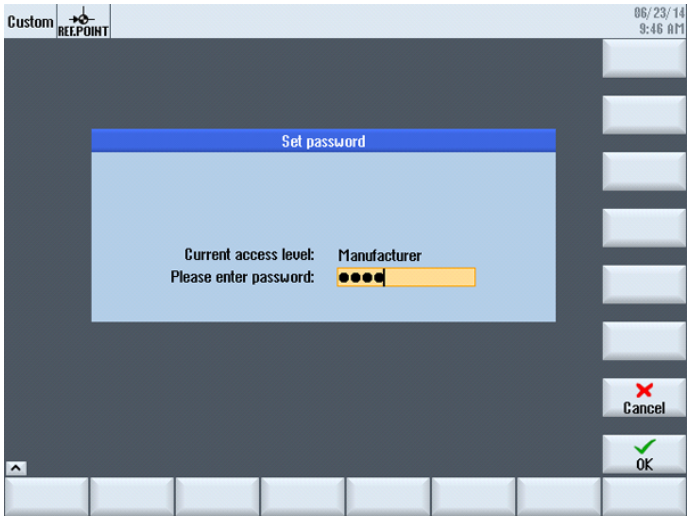



Figure 4-5 Set password dialog

- Change password

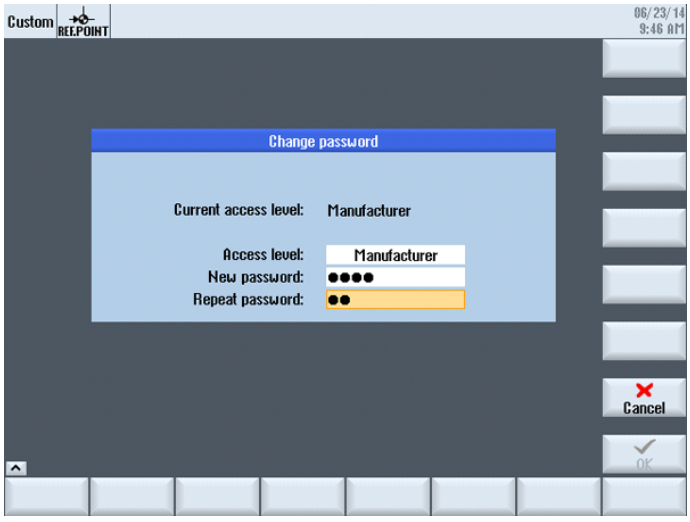



Figure 4-6 Change password dialog

- Delete password

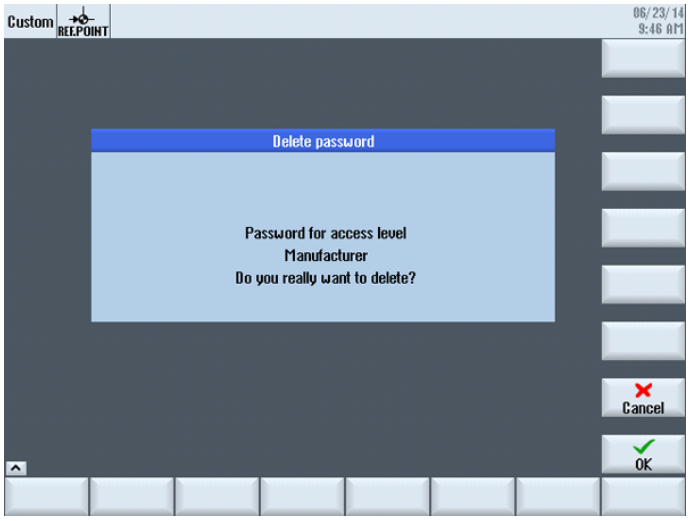



Figure 4-7 Delete password dialog

**Note**

Password functionality is provided with these dialogs. The dialogs do not correspond to the dialogs of SINUMERIK Operate.

The dialogs can be called either via the Load softkey (LS) function or the Load mask (LM) function:

1. Call via the Load softkey (LS) function:
   `LS("Passwd","slespasswd.com",1)`
   Extension of the vertical softkeys:
   - Softkey 1: Set password
   - Softkey 2: Change password
   - Softkey 3: Delete password
2. Alternatively, the three masks can be directly jumped to by calling the (LM) function:
   - Set password: `LM("PWD_SET", "slespasswd.com",1)`
   - Change password: `LM("PWD_CHG", "slespasswd.com",1)`
   - Delete password: `LM("PWD_CLEAR", "slespasswd.com",1)`

## 4.1.6 Using display images/graphics

### Use of graphics

A distinction is made between:

- Display images / graphics in the graphic area.
- Help displays illustrating, for example, individual variables, which are superimposed in the graphic area.
- More Help displays can be configured instead of short text or an input/output field, which you position where you like.

### Storage locations

The image matching the resolution of the connected monitor is searched for in the associated resolution directories in the following sequence:

*[System user directory]*/ico/ico<Resolution>
*[System oem directory]*/ico/ico<Resolution>
*[System add_on directory]*/ico/ico<Resolution>

If the image is not displayed or not found, copy it into one of the following directories for a resolution of 640 x 480 pixels:

*[System user directory]*/ico/ico640
*[System oem directory]*/ico/ico640
*[System add_on directory]*/ico/ico640

**Note**

With the different panel resolutions, the images are positioned proportionally.

# 4.2 Defining softkey menus

## Definition

The term softkey menu is used to refer to all the horizontal and vertical softkeys displayed on a mask. In addition to the existing softkey menus, it is possible to define other menus, which partially or completely overwrite the existing menus.

The names of the softkeys are predefined. Not all softkeys need to be assigned.

HSx x 1 - 8, Horizontal softkeys 1 to 8

VSy y 1 - 8, Vertical softkeys 1 to 8

The definition of a softkey menu (softkey menu definition block) is basically structured as follows:

| Definition block | Comment | Section reference |
|---|---|---|
| `//S...` | ;Start identifier of softkey menu | |
| `HSx=...` | ;Define softkeys | |
| `PRESS(HSx)`<br>`  LM...`<br>`END_PRESS` | ;Method start identifier<br>;Actions<br>;Method end identifier | See Chapter Methods (Page 126) |
| `//END` | ;End identifier of softkey menu | |

## Description

Properties are assigned to softkeys during definition of the softkey menu.

## Programming

| | | |
|---|---|---|
| Syntax: | **//S***(Identifier)*<br>...<br>//END | ;Start identifier of softkey menu<br><br>;End identifier of softkey menu |
| | See also Section Extended configuration syntax (Page 43). | |
| Description: | Defines the softkey menu | |
| Parameter: | Identifier | Name of the softkey menu |
| | Text or image file name | |
| Syntax: | **SK** = *(Text[, access level][, Status][, Alignment of the softkey image][, text alignment referred to the softkey image])* | |
| Description: | Define softkey | |
| Parameter: | SK | Softkey, e.g. HS1 to HS8, VS1 to VS8 |

| | | | |
|---|---|---|---|
| | Text | Enter text | |
| | Image file name | "\\my_pic.png"<br>or via separate text file $85199, e.g. with the following text in the (language-specific) text file: 85100 0 0 "\ \my_pic.png".<br>The image size which can be displayed on a softkey depends on the OP used: | |
| | | OP 08: | 640 x 480 mm → 25 x 25 pixels |
| | | OP 010: | 640 x 480 mm → 25 x 25 pixels |
| | | OP 012: | 800 X 600 mm → 30 x 30 pixels |
| | | OP 015: | 1024 X 768 mm → 40 x 40 pixels |
| | | OP 019: | 1280 x 1024 mm → 72 x 72 pixels |
| | Access level | ac0 to ac7 (ac7: default) | |
| | Status | se1: visible (default)<br>se2: disabled (gray text)<br>se3: displayed (last softkey used) | |
| | Alignment of the softkey image | PA (PictureAlignment) | |
| | | Valid values:<br>0: Left<br>1: Right<br>2: Centered<br>3: Top (default)<br>4: Bottom | |
| | Text alignment in relation to the softkey image | TP (TextAlignedToPicture) | |
| | | Valid values:<br>0: Text is not aligned to the image<br>1: Text is aligned to the image (default) | |

**Note**

Enter %n in the softkey text to create a line break.

A maximum of 2 lines with 9 characters each are available.

## Assigning access level

Operators can only access information on this access level and lower access levels (see also List of access levels (Page 308)).

## Example

```
//S(Menu1)                                ; Start identifier of softkey menu
HS1=("NEW", ac6, se2)                     ; Define softkey HS1, assign the label "NEW", protec-
                                          tion level 6, and the status "disabled"
```

```
HS2=("\\image1.png")                          ; Assign a softkey to the graphic
HS3=("Exit")
HS4=(["Confirm","\\sk_ok.png"],PA0,TP1)       ; Softkey with text and graphic, Text="Confirm", Im-
                                              age="sk_ok.png", Alignment of the softkey image: Left,
                                              text is aligned to the image


VS1=("sub mask")
VS2=($85011, ac7, se2)                        ; Define softkey VS2, assign the text from the lan-
                                              guage file, protection level 1, and the status "disa-
                                              bled"
VS3=("Cancel", ac1, se3)                      ; Define softkey VS3, assign the label "Cancel", pro-
                                              tection level 1 and the status "highlighted"
VS4=("OK", ac6, se1)                          ; Define softkey VS4, assign the label "OK", protec-
                                              tion level 6 and the status "visible"
VS5=(SOFTKEY_CANCEL,,se1)                     ; Define cancel standard softkey VS5 and assign the
                                              status "visible"
VS6=(SOFTKEY_OK,,se1)                         ; Define OK standard softkey VS6 and assign the status
                                              "visible"
VS7=(["\\image1.png","OEM text"],,se1)        ; Define softkey VS7, assign an image, assign the la-
                                              bel "OEM Text" and the status "visible"
VS8=(["\\image1.png",$83533],,se1)            ; Define softkey VS8, assign an image, assign text
                                              from language file and the status "visible"


PRESS(HS1)                                    ; Method start identifier
  HS1.st=" Calculate"                         ; Assign a label text to the softkey
...
END_PRESS                                     ; Method end identifier

PRESS(RECALL)                                 ; Method start identifier
  LM("Maske21")                               ; Load dialog
END_PRESS                                     ; Method end identifier
//END                                         ; End identifier of softkey menu
```

## 4.2.1 Changing softkey properties during runtime

### Description

The softkey properties Text, Access Level and Status can be changed in the methods during runtime.

## Programming

| | | | |
|---|---|---|---|
| Syntax: | SK.st = "Text"<br>SK.ac = Access level<br>SK.se = Status<br>SK.pa = "Alignment of the softkey image"<br>SK.tp = "Text alignment in relation to the softkey image" | | ;Softkey with label<br>;Softkey with protection level<br>; Softkey with status<br>;Softkey with image<br>;Softkey with image and label |
| Description: | Assign properties | | |
| Parameter: | Text | | Label text in inverted commas |
| | Access level | | Range of values: 0 ... 7 |
| | Status | 1:<br>2:<br>3: | Visible and operator-controllable<br>Disabled (gray text)<br>Highlighted (last softkey used) |
| | Alignment of the softkey image | 0:<br>1:<br>2:<br>3:<br>4: | Left<br>Right<br>Centered<br>Top (default)<br>Bottom |
| | Text alignment in relation to the softkey image | 0:<br>1: | Text is not aligned to the image<br>Text is aligned to the image (default) |

## Example

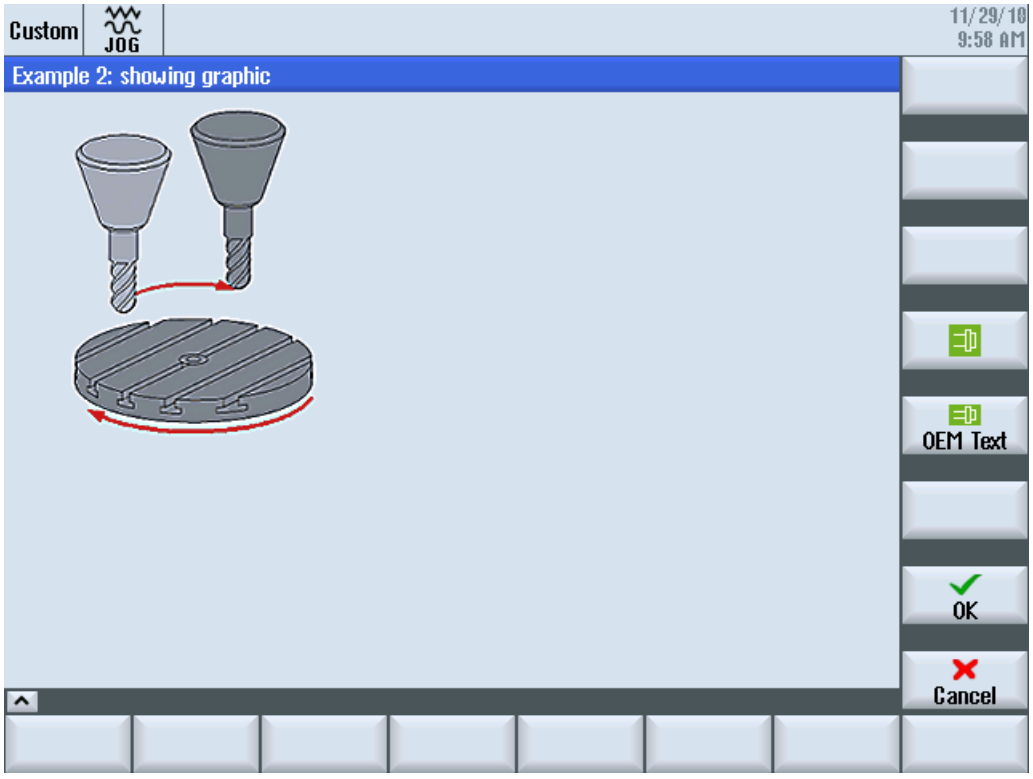



Figure 4-8 Example 3: Graphics and softkeys

```
//S(Start)
HS7=("Example", ac7, se1)

PRESS(HS7)
  LM("Maske3")
END_PRESS

//END

//M(Maske3/"Example 2: showing graphic"/"example.png")
HS1=("")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")
VS2=("")
```

```
VS3=("")
VS4=("\\sp_ok.png",,SE1)
VS5=(["\\sp_ok_small.png","OEM Text"],,SE1)
VS6=("")
VS7=(SOFTKEY_OK,,SE1)
VS8=(SOFTKEY_CANCEL,,SE1)
PRESS(VS4)
  EXIT
END_PRESS
PRESS(VS5)
  EXIT
END_PRESS
PRESS(VS7)
  EXIT
END_PRESS
PRESS(VS8)
  EXIT
END_PRESS
//END
```

## 4.2.2 Language-dependent text

### Overview

Language-dependent texts are used for:

- Softkey labels
- Headings
- Help texts
- Any other texts

The language-dependent texts for dialogs are stored in text files.

The text files are stored in the following directories:

- *[System user directory]*\lng
- *[System oem directory]*\lng
- *[System add_on directory]*\lng

**Note**

The text files must be saved in the same way as the project files.

For example:

*[System user directory]*/lng/[Text file]

*[System user directory]*/proj/[Configuration file]

| | |
|---|---|
| alsc.txt | Language-dependent texts for the Siemens standard cycles |
| almc.txt | Language-dependent texts for the Siemens measuring cycles |

The text files used during program runtime are specified in the "easyscreen.ini" file:

```
[LANGUAGEFILES]
LngFile03 = user.txt      ;->user<_xxx>.txt (e.g.: user_eng.txt)
```

In this instance, the "user.txt" file has been chosen as an example of a text file. The name can always be freely selected. Depending on the language of the texts within the file, the relevant language code must be added using the following syntax. An underscore followed by the relevant language identifier is added after the name e.g. "user_eng.txt".

**Note**

You must not use LngFile01 and LngFile02 for user texts because they are already assigned in the standard "easyscreen.ini".

## See also

List of language codes used in file names (Page 310)

## Mask-specific language files

In order to avoid overlaps for the number ranges used in the language files, only certain language files can be used in a specific mask.

This is the reason that the language files to be used are listed in the mask definition line, separated by a comma. The file listed at the last location has the highest priority (analogous to the logic in "easyscreen.ini").

All of the language-dependent texts used in the mask are predominantly searched for in these language files. If the text is not found in these, then a search is made in text files configured in "easyscreen.ini".

If no language files have been defined in the mask, then a search is only made in the language files specified in "easyscreen.ini".

**Note**

The English language file is used (default) if a language file is not available in the currently selected language.

**Examples:**

```
//M(MyMask/$85597///// //"mytexts.txt, general.txt, mask.txt")
DEF ….
```

You can also use this procedure for the start softkeys:

```
//S(Start, "mytexts.txt, general.txt, mask.txt")
VS1=($85597)
PRESS(VS1)
   LM("MyMask","masks.com")
END_PRESS
```

## Format of text files

The text files must be saved in UTF-8 format.

**Note**

Make sure that the coding is set to UTF-8 in the editor that you are using when saving the configuration and language files.

## Format of a text entry

| | | | |
|---|---|---|---|
| Syntax: | 8xxxx 0 0 "Text" | | |
| Description: | Assignment between text number and text in the file | | |
| Parameter: | xxxx | 85,000 to 89,999<br>900,000 to 999,999 | Text identification number range reserved for users. You must assign unique numbers. |
| | "Text" | | Text that appears in the dialog |
| | %n | | Control characters in the text for creating a line break |

Parameters 2 and 3 are separated by blanks and act as control characters for alarm text output. To ensure that the text format is identical to that of the alarm texts, you must set zero.

**Examples of language-dependent texts:**

```
85000 0 0 "Retraction plane"
85001 0 0 "Drilling depth"
85002 0 0 "Pitch"
85003 0 0 "Pocket radius"
```

# 4.3 Configuring the online help

## 4.3.1 Overview

In addition to the existing extensive online help, you also have the option of generating a manufacturer-specific online help and then linking this into SINUMERIK Operate.

This online help is generated in the HTML format, i.e. it comprises HTML documents that are linked with one another. The subject being searched for is called in a separate window from a contents or index directory. Similar to a document browser (e.g. Windows Explorer), a list of possible selections is displayed in the left-hand half of the window and when you click on the required subject, the explanation is displayed in the right hand half of the window.

Context sensitive selection of online help pages is not possible.

### Procedure

1. Generating HTML files
2. Generating a help book
3. Integrating the online help in SINUMERIK Operate
4. Saving help files

### Other application cases

Online help for the following OEM-specific expansions can be created and used to supplement the SINUMERIK Operate online help system:

- Online help for **cycles and/or M functions** of the machine manufacturer which extend the programming options for SINUMERIK control systems. This online help is called in just the same way as the SINUMERIK Operate online help "Programming".
- Online help for OEM-specific **variables** of the machine manufacturer. This online help is called from the variable view of SINUMERIK Operate.

### Programming online help

You can use the "SINUMERIK HMI programming package sl" for additional options for configuring the online help. Using this programming package, it is possible to develop high-level language applications in the C++ programming language for SINUMERIK Operate on the NCU 7x0.

**Note**

The "SINUMERIK HMI programming package sl" must be ordered separately as a software option. The associated documentation is provided together with the programming package.

## 4.3.2 Generating HTML files

Generating help files in the HTML format. It is possible to save all information in a single HTML file or to distribute the information over several HTML files.

You can assign the file names yourself, however, you must observe the following:

- References within HTML files should always be specified with relative paths. Only then can it be ensured that the references function in precisely the same way on both the development computer as well as on the target system.
- If jumps are to be made to certain points within an HTML file per link, then so-called anchors must be defined for this purpose.
  Example of an HTML anchor:
  `<a name="myAnchor">This is an anchor</a>`
- The contents of HTML documents must be saved with the UTF-8 coding. Only then is it guaranteed that the HTML documents are correctly displayed in all of the country languages supported by SINUMERIK Operate.
- The following sub-sets of the HTML functional scope are supported:

### HTML tags

| Tag | Description | Comment |
|---|---|---|
| a | Anchor or link | Supported attributes: href and name |
| address | Address | |
| big | Larger font | |
| block quote | Indented paragraph | |
| body | Document body | Supported attributes: bgcolor (#RRGGBB) |
| br | Line break | |
| center | Centered paragraph | |
| cite | Inline citation | Same effect as tag i |
| code | Code | Same effect as tag tt |
| dd | Definition data | |
| dfn | Definition | Same effect as tag i |
| div | Document division | The standard block attributes are supported |
| dl | Definition list | The standard block attributes are supported |
| dt | Definition term | The standard block attributes are supported |
| em | Emphasized | Same effect as tag i |
| font | Font size, family, color | Supported attributes: size, face, and color (#RRGGBB) |
| h1 | Level 1 heading | The standard block attributes are supported |
| h2 | Level 2 heading | The standard block attributes are supported |
| h3 | Level 3 heading | The standard block attributes are supported |
| h4 | Level 4 heading | The standard block attributes are supported |
| h5 | Level 5 heading | The standard block attributes are supported |
| h6 | Level 6 heading | The standard block attributes are supported |
| head | Document header | |

| Tag | Description | Comment |
|---|---|---|
| hr | Horizontal line | Supported attributes: width (can be specified as absolute or relative value) |
| html | HTML document | |
| i | Italic | |
| img | Image | Supported attributes: src, width, height |
| kbd | User-entered text | |
| meta | Meta information | |
| li | List item | |
| nobr | Non-breakable text | |
| ol | Ordered list | The standard attributes for lists are supported |
| p | Paragraph | The standard block attributes are supported (default setting: left-aligned) |
| pre | Preformatted text | |
| s | Strikethrough | |
| samp | Sample code | Same effect as tag tt |
| small | Small font | |
| span | Grouped elements | |
| strong | Strong | Continuous text is heavily emphasized, replaces tag b |
| sub | Subscript | |
| sup | Superscript | |
| table | Table | Supported attributes: border, bgcolor (#RRGGBB), cellspacing, cellpadding, width (absolute or relative), height |
| tbody | Table body | No effect |
| td | Table data cell | The standard attributes for table cells are supported |
| tfoot | Table footer | No effect |
| th | Table header cell | The standard attributes for table cells are supported |
| thead | Table header | This is used to print tables that extend over several pages |
| title | Document title | |
| tr | Table row | Supported attributes: bgcolor (#RRGGBB) |
| tt | Typewrite font | |
| u | Underlined | |
| ul | Unordered list | The standard attributes for lists are supported |
| var | Variable | Same effect as tag tt |

## Block attributes

The following attributes are supported by the tags div, dl, dt, h1, h2, h3, h4, h5, h6, p:

- align (left, right, center, justify)
- dir (ltr, rtl)

### Standard attributes for lists

The following attributes are supported by tags ol and ul:

- type (1, a, A, square, disc, circle)

### Standard attributes for tables

The following attributes are supported by tags td and th:

- width (absolute, relative, no-value)
- bgcolor (#RRGGBB)
- colspan
- rowspan
- align (left, right, center, justify)
- valign (top, middle, bottom)

### CSS properties

The following table includes the supported CSS functional scope:

| Property | Values | Description |
|---|---|---|
| background color | <color> | Background color for elements |
| background-image | <uri> | Background image for elements |
| color | <color> | Foreground color for text |
| text-indent | <length>px | Indent the first line of a paragraph in pixels |
| white-space | normal \| pre \| nowrap \| pre-wrap | Defines how whitespace characters are handled in HTML documents |
| margin-top | <length>px | Width of the upper edge of the paragraph in pixels |
| margin-bottom | <length>px | Width of the lower edge of the paragraph in pixels |
| margin-left | <length>px | Length of the left hand edge of the paragraph in pixels |
| margin-right | <length>px | Width of the right-hand edge of the paragraph in pixels |
| vertical-align | baseline \| sub \| super \| middle \| top \| bottom | Vertical alignment for text (in tables, only the values middle, top and bottom are supported) |
| border-color | <color> | Border color for text tables |
| border-style | none \| dotted \| dashed \| dot-dash \| dot-dot-dash \| solid \| double \| groove \| ridge \| inset \| outset | Border style for text tables |
| background | [ <'background-color'> \|\| <'background-image'> ] | Short notation for background property |
| page-break-before | [ auto \| always ] | Page break before a paragraph/table |
| page-break-after | [ auto \| always ] | Page break after a paragraph/table |
| background-image | <uri> | Background image for elements |

### Supported CSS selectors

All of the CSS 2.1 selector classes are supported, with the exception of so-called pseudo-selector classes, such as :first-child, :visited and :hover.

## 4.3.3 Generating the help book

The help book is an XML file in which the structure of the online help is defined. In this file, you define:

- HTML documents
- Contents and subject index

### Syntax for the help book

| Tag | Number | Meaning | |
|---|---|---|---|
| HMI_SL_HELP | 1 | Root element of the XML document | |
| I-BOOK<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I<br>I | + | Identifies a help book. The name can be freely selected under the constraint that no name predefined by the system is used (such as sinumerik_alarm_plc_pmc).<br>In the example, the name of the help book is "hmi_myhelp"<br>Attributes: | |
| | | ref | Identifies the HTML document that is displayed as the entry page for the help book. |
| | | title | Title of the help book that is displayed in the table of contents. |
| | | helpdir | Directory that contains the online help of the help book. |
| I-ENTRY<br>II<br>II<br>II<br>II<br>II<br>II<br>II<br>II | * | Section of the online help<br>Attributes: | |
| | | ref | Identifies the HTML document that is displayed as entry page for the section. |
| | | title | Title of the section that is displayed in the table of contents. |
| II-INDEX_ENTRY<br>II<br>II<br>II<br>II<br>II<br>II<br>II<br>II | * | Subject (keyword) to be displayed<br>Attributes: | |
| | | ref | Identifies the HTML document that is jumped to for this subject index entry. |
| | | title | Title of the subject that is displayed in the subject index. |

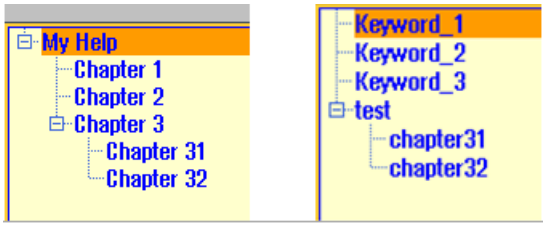
The following applies for the "Number" column:
* means 0 or more
+ means 1 or more

## Example for a help book

In the following example, the structure of a help book with the "My Help" name is described. Further, it forms the basis for the table of contents and subject index.

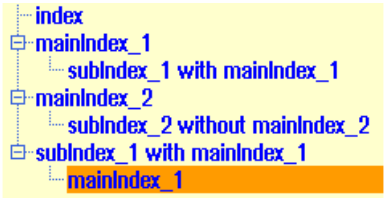
```
<?xml version="1.0" encoding="utf-8"?>
<HMI_SL_HELP language="en-US">
  <BOOK ref="index.html" title="My Help" helpdir="hmi_myhelp">
    <ENTRY ref="section_1.html" title="Section 1">
      <INDEX_ENTRY ref="section_1html#Keyword_1" title="Keyword_1"/>
      <INDEX_ENTRY ref="section_1.html#Keyword_2" title="Keyword_2"/>
    </ENTRY>
    <ENTRY ref="section_2.html" title="Section 2">
      <INDEX_ENTRY ref="section_2.html#Keyword_3" title="Keyword_3"/>
    </ENTRY>
    <ENTRY ref="section_3.html" title="Section 3">
      <ENTRY ref="section_31.html" title="Section 31">
        INDEX_ENTRY ref="section_31.html#test" title="test;section31"/>
      </ENTRY>
      <ENTRY ref="section_32.html" title="Section 32">
        INDEX_ENTRY ref="section_32.html#test" title="test;section32"/>
      </ENTRY>
    </ENTRY>
  </BOOK>
</HMI_SL_HELP>
```

The book comprises three sections, whereby the third section has two subsections. The various subject words (keywords) are defined within the section.



You have the following three options to format the subject index:

1. Single entry:
   `<INDEX_ENTRY ...title="index"/>`
2. Two two-stage entry, whereby each title has a main and a subentry. Separate the entries from one another using a comma.
   `<INDEX_ENTRY ...title="mainIndex_1,subIndex_1 with mainIndex_1"/>`
3. Two-stage entry, whereby the first title is the main entry and the second title is the subentry. Separate the entries from one another using a semicolon.
   `<INDEX_ENTRY ...title="mainIndex_2;subIndex_2 without mainIndex_1"/>`




## 4.3.4 Integrating the online help in SINUMERIK Operate

If you wish to integrate the generated help book into the online help system of SINUMERIK Operate, then you require the "slhlp.xml" file.

### Format description of the "slhlp.xml"

| Tag | Number | Meaning | |
|---|---|---|---|
| CONFIGURATION | 1 | Root element of the XML document. Designates that this involves a configuration file. | |
| I-OnlineHelpFiles | 1 | Introduces the section about the online help books. | |
| II-<help_book> | * | Introduces the section of a help book. | |
| III-EntriesFile<br>III<br>III<br>III<br>III<br>III | 1 | File name of the help book with the list of contents and subject (keyword) entries.<br>Attributes: | |
| | | value | Name of the XML file |
| | | type | Data type of the value (QString) |
| III-Technology<br>III<br>III<br>III<br>III<br>III<br>III<br>III<br>III<br>III<br>III<br>IIII | 0, 1 | Specifies the technology that applies to the help book.<br>"All" applies to all technologies.<br>If the help book applies to several technologies, then the technologies are listed separated by comma.<br>Possible values:<br>All, Universal, Milling, Turning, Grinding, Stroking, Punching<br>Attributes: | |
| | | value | Technology data |
| | | type | Data type of the value (QString) |

| Tag | Num-ber | Meaning | |
|---|---|---|---|
| III -DisableSearch<br>III<br>III<br>III<br>III<br>III | 0, 1 | Disable the subject (keyword) search for the help book.<br>Attributes: | |
| | | value | true, false |
| | | type | type, data type of the value (bool) |
| III-DisableFullTextSearch<br>III<br>III<br>III<br>III | 0, 1 | Disable the full text search for the help book.<br>Attributes: | |
| | | value | true, false |
| | | type | type, data type of the value (bool) |
| III-DisableIndex<br>III<br>III<br>III<br>III | 0, 1 | Disable the subject index for the help book.<br>Attributes: | |
| | | value | true, false |
| | | type | type, data type of the value (bool) |
| III-DisableContent<br>III<br>III<br>III<br>III | 0, 1 | Disable the table of contents for the help book.<br>Attributes: | |
| | | value | true, false |
| | | type | type, data type of the value (bool) |
| III-DefaultLanguage<br>III<br>III<br>III<br>III<br>III | 0, 1 | Abbreviation for the language that should be displayed if the actual country language is available for the help book.<br>Attributes: | |
| | | value | chs, deu, eng, esp, fra, ita, ... |
| | | type | Data type of the value (QString) |

The following applies for the "Number" column:
* means 0 or more

## Example of a file "slhlp.xml"

The help book "hmi_myhelp.xml" is made known to SINUMERIK Operate in the following example.

The subject index has not been activated for the help book.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE CONFIGURATION>
<CONFIGURATION>
   <OnlineHelpFiles>
     <hmi_myHelp>
       <EntriesFile value="hmi_myhelp.xml" type="QString"/>
       <DisableIndex value="frue" type="bool"/>
     </hmi_myHelp>
   </OnlineHelpFiles>
</CONFIGURATION>
```

## 4.3.5 Saving help files

### Saving help files in the target system

1. Open the /**oem**/sinumerik/hmi/hlp directory and create a new folder for the required language. For this purpose, use the specified language code.
   It is mandatory that the folder names are written in lower-case letters.
   For instance, if you are integrating a help function for German and English, then create the "deu" and "eng" folders.
2. Place the help book, e.g. "hmi_myhelp.xml" in the "deu" and "eng" folders.
3. Copy the help files into the directories, e.g. /**oem**/sinumerik/hmi/hlp/deu/hmi_myhelp for German and /**oem**/sinumerik/hmi/hlp/eng/hmi_myhelp for English help files.
4. Place the configuration file "slhlp.xml" into the directory /**oem**/sinumerik/hmi/cfg.
5. Restart the HMI.

**Note**

When displaying the list of contents and subject index of a help book, the help files are saved in the binary format (slhlp_<Hilfe-Buch_*.hmi) under the directory /siemens/sinumerik/sys_cache/hmi/hlp for faster use. If you change the help book, you must always delete these files.

## 4.3.6 Help files in PDF format

In addition to help files in HTML format, you can also include PDF-format information in the operating software. Individual PDF helps can be opened with links from the table of contents or index, or directly from HTML files.

### Storing PDF helps

Copy the PDF helps to one of the following directories:

```
/oem/sinumerik/hmi/hlp/<lng>/<hmi_myhelp>
/user/sinumerik/hmi/hlp/<lng>/<hmi_myhelp>
```

### Including PDF helps

Bind in dialog configurations or configurations of table of contents and indexes with the extension "pdf" in the same manner as for the "html" extension:

```
<ENTRY ref="myFile.pdf" title="Help 1">
```

Create a link from HTML files to the PDF help:

```
<a href="myFile.pdf">My Help File</a>
```

**Note**

It is not possible to select context-sensitive jump labels in the PDF help, select jump labels or make jumps to other HTML or PDF files.

The search function is possible only within a PDF file. A higher-level search over multiple PDF helps is not supported.

# Variables

# 5

## 5.1 Defining variables

### Variable value

The major property of a variable is its value.

The value of variables can be assigned by means of:

- Default settings when defining variables
- Assignment to a system or user variable
- A method

### Programming

| | | |
|---|---|---|
| Syntax: | Identifier.**val** = Variable value<br>Identifier = Variable value | |
| Description: | Variable value val (value) | |
| Parameter: | Identifier: | Name of the variable |
| | Variable value: | Value of the variable |
| Example: | `VAR3 = VAR4 + SIN(VAR5)`<br>`VAR3.VAL = VAR4 + SIN(VAR5)` | |

### Variable status

The "Variable status" property scans a variable for valid content during runtime. This property can be read and written with the value FALSE = 0.

### Programming

| | | |
|---|---|---|
| Syntax: | Identifier.**vld** | |
| Description: | Variable status vld (validation) | |
| Parameter: | Identifier: | Name of the variable |
| | | The result of the scan can be: |
| | FALSE = | invalid value |
| | TRUE = | valid value |
| Example: | `IF VAR1.VLD == FALSE`<br>`  VAR1 = 84`<br>`ENDIF` | |

# 5.2 Application examples

## Help variables

Help variables are internal arithmetic variables. Arithmetic variables are defined like other variables, but have no other properties apart from variable value and status, i.e. Help variables are not visible in the dialog. Help variables are of the VARIANT type.

## Programming

| | | |
|---|---|---|
| Syntax: | DEF *Identifier* | |
| Description: | Internal arithmetic variables of the VARIANT type | |
| Parameter: | Identifier: | Name of the Help variable |
| | | |
| Example: | `DEF OTTO ;Definition of a Help variable` | |

| | | |
|---|---|---|
| Syntax: | Identifier.val = *Help variable value*<br>Identifier = *Help variable value* | |
| Description: | A value is assigned to a Help variable in a method. | |
| Parameter: | Identifier: | Name of the Help variable |
| | Help variable value: | Content of the Help variables |

**Example:**

```
LOAD
  OTTO = "Test"             ; The value "Test"  is assigned to help variable Otto.
END_LOAD
LOAD
  OTTO = REG[9].VAL         ; The value of the register is assigned to help variable
END_LOAD                    Otto
```

## Calculation with variables

Variables are calculated every time you exit an IO field (by pressing the ENTER or TOGGLE key). The calculation is configured in a CHANGE method that is processed every time the value changes.

You can scan the variable status to ascertain the validity of the value of the variable, e.g.

```
IF VAR1.VLD == FALSE
   VAR1 = 84
ENDIF
```

### Addressing system variables indirectly

A system variable can also be addressed indirectly, i.e. as a function of another variable:

```
PRESS(HS1)
  ACHSE=ACHSE+1
  WEG.VAR="$AA_DTBW["<<ACHSE<<"]"       ;Address axis address via variable
END_PRESS
```

### Changing softkey labels

Example:

```
HS3.st = "New text"                     ;Change softkey label
```

# 5.3 Example 1: Assigning the variable type, texts, help display, colors, tooltips

## Example 1a

The following example defines a variable for which the properties variable type, texts, help display and colors are set.

| DEF Var1 = (R///,"Actual value",,"mm"//"Var1.png"////8,2) | | | |
|---|---|---|---|
| | Variable type: | | REAL |
| | Texts: | | |
| | | Short text: | Actual value |
| | | Unit text: | mm |
| | Help screen: | | Var1.png |
| | Colors: | | |
| | | Foreground color: | 8 (brown) |
| | | Background color: | 2 (orange) |

## Example 1b

The following example defines a variable for which the properties variable type, default setting, texts, tooltip, input mode and position of short text are set.

| DEF Var2 = (I//5/"","Value","","","" Tooltip text"/wr2///20,250,50) | | | |
|---|---|---|---|
| | Variable type: | | INTEGER |
| | Default setting: | | 5 |
| | Texts: | | |
| | | Short text: | Value (possible language text ID) |
| | | Tooltip: | ToolTipText |
| | Attributes: | | |
| | | Input mode: | Reading and writing |
| | Position of short text: | | |
| | | Distance from left: | 20 |
| | | Distance from top: | 250 |
| | | Width: | 50 |

## See also

Variable parameters (Page 92)

# 5.4 Example 2: Assigning the Variable Type, Limits, Attributes, Short Text Position properties

## Example 2

The following example defines a variable for which the properties variable type, limits, input mode, alignment and position are set.

| DEF Var2 = (I/0,10///wr1,al1///,,300) | | | |
|---|---|---|---|
| | Variable type: | | INTEGER |
| | Limits or toggle field entries: | | MIN: 0<br>MAX: 10 |
| | Attributes: | | |
| | | Input mode: | Read-only |
| | | Alignment of short text: | Right-justified |
| | Position of short text: | | |
| | | Width: | 300 |

## See also

Variable parameters (Page 92)

# 5.5 Example 3: Assigning the Variable Type, Default, System or User Variable, Input/Output Field Position properties

## Example 3

The following example defines a variable for which the properties variable type, default setting, system or user variable and position are set.

| DEF Var3 = (R//10////"$R[1]"//300,10,200//) | | | |
|---|---|---|---|
| | Variable type: | | REAL |
| | Default setting: | | 10 |
| | System or user variable: | | $R[1] (R-Parameter 1) |
| | Position of short text: | | Default position in relation to input/output field |
| | Position of input/output field: | | |
| | | Distance from left: | 300 |
| | | Distance from top: | 10 |
| | | Width: | 200 |

## See also

Variable parameters (Page 92)

# 5.6 Example 4: Toggle field and list field

## Example 4a

Various entries in the toggle field:

```
DEF Var1 = (I/* 0,1,2,3)                ;Simple toggle field to toggle numeric values
DEF Var2 = (S/* "On", "Off")            ;Simple toggle field to toggle strings
DEF Var3 = (B/* 1="On", 0="Off")        ;Extended toggle field to toggle numeric values whereby a
                                         display text is assigned to each numeric value
DEF Var4 = (R/* ARR1)                   ;Toggle field, which obtains its values to be toggled from
                                         an array
```

## Example 4b

The list field corresponds to the configuration of a toggle field, but the display type for the list field (variable attribute DT = 4) must also be set.

| DEF VAR_LISTBOX_Text = (S/*$80000,$80001,$80002,$80003,$80004/$80001//DT4////200,,340,60) | | |
|---|---|---|
| Variable type: | | STRING |
| Limits / toggle field: | | *$80000,$80001,$80002,$80003,$80004 (list of the language-dependent texts to be displayed) |
| Default setting: | | $80001 |
| Attributes: | | |
| | Display type: | 4 (list field) |
| Position of input/output field: | | |
| | Distance from left: | 200 |
| | Width: | 340 |
| | Height: | 60 |
| Colors: | | |
| | Foreground color: | 6 (blue) |
| | Background color: | 10 (white) |

# 5.7 Example 5: Image display

## Example 5

Displaying an image instead of a short text: The size and position of the image is defined under "Position of input/output field (left, top, width, height)".

| **DEF VAR6= (V///,"\\image1.png" ////160,40,50,50)** | | | |
|---|---|---|---|
| | Variable type: | | VARIANT |
| | Texts: | | |
| | | Short text: | image1.png |
| | Position of input/output field: | | |
| | | Distance from left: | 160 |
| | | Distance from the top: | 40 |
| | | Width: | 50 |
| | | Height: | 50 |

# 5.8 Example 6: Progress bar

The progress bar is a special display type of the input/output field and is designed for display without input.

Basically there are two progress bar types:

1. Progress bars with up to two color changes, e.g. for temperature or utilization display (see example 6a)
2. Progress bars to display progress (no color change) in the Operate style (see example 6b)

## Example 6a

Progress bar with two color changes:



Figure 5-1 Progress bar with two color changes

| DEF PROGGY0 = (R/0,150,50,100///DT1,DO0//"$R[10]"//,,150/3,4,,,,,,,9,7) | | |
|---|---|---|
| Variable type: | | REAL |
| Limits / toggle field: | | |
| | MIN: | 0 |
| | MAX: | 150 |
| | Signal value SVAL1: | 50 |
| | Signal value SVAL2: | 100 |
| Attributes: | | |
| | Display mode DT: | 1 (progress bar) |
| | Display option DO: | 0 (from left to right (default)) |
| System or user variable: | | $R[10] |
| Position of input/output field: | | |
| | Width: | 150 |
| Colors: | | |
| | Foreground color: | 3 (dark green) |
| | Background color: | 4 (light gray) |
| | Signal color SC1: | 9 (yellow) |
| | Signal color SC2: | 7 (red) |

To use a progress bar with color change, the display mode DT (DisplayType) must be set to 1.

The orientation of the progress bar is determined via the attribute display option DO (DisplayOption):

0: From left to right (default)

1: From right to left

2: From bottom to top

3: From top to bottom

A MIN and a MAX value must be specified for the display of the progress bar (in the example MIN: 0, MAX: 150).

Depending on the current value of the variable PROGGY0, this setting already displays a progress bar with the foreground color 3 (= dark green) and the background color 4 (= light gray).

Optionally, one or two signal values SVAL1 and SVAL2 (limit parameters) can be defined (in the example SVAL1: 50 and SVAL2: 100). With these signal values, the foreground color of the progress bar changes. The appropriate signal colors are specified via parameters SC1 and SC2 (in the above example SC1: 9 (= yellow) and SC2: 7 (=red)).

The following rule applies when specifying limit values for the progress display:

MIN < SVAL1 < SVAL2 < MAX.

## Example 6b

Progress bar without color change:



Figure 5-2 Progress bar

| DEF PROGGY0 = (R/0,150///DT2,DO0//"$R[10]"//,,150/6,10) | | | |
|---|---|---|---|
| | Variable type: | | REAL |
| | Limits / toggle field: | | |
| | | MIN: | 0 |
| | | MAX: | 150 |
| | Attributes: | | |
| | | Display type DT: | 2 (progress bar) |
| | | Display option DO: | 0 (from left to right (default)) |
| | System or user variable: | | $R[10] |
| | Position of input/output field: | | |
| | | Width: | 150 |
| | Colors: | | |
| | | Foreground color: | 6 (blue) |
| | | Background color: | 10 (white) |

To use a progress bar without color change, the display mode DT (DisplayType) must be set to 2.

The orientation of the progress bar is determined via the attribute display option DO (DisplayOption) (see description for example 6a).

A MIN and a MAX value must be specified for the display of the progress bar (in the example MIN: 0, MAX: 150).

Depending on the current value of the variable PROGGY0, this setting displays a progress bar with the foreground color 6 (= blue) and the background color 10 (= white).

# 5.9 Example 7: Password input mode (asterisk)

## Example 7

To implement a field with hidden input, e.g. to enter a password, then display mode DT must be set to 5. Asterisks are then displayed instead of the entered characters.



Figure 5-3 Password input mode (asterisk)

| DEF VAR_SET_PWD=(S//""//DT5) | | | |
|---|---|---|---|
| | Variable type: | | STRING |
| | Default setting: | | Empty string |
| | Attributes: | | |
| | | Display mode DT: | 5 (password input mode) |

# 5.10 Variable parameters

## Parameter overview

The following overview provides a brief explanation of the variable parameters. Subsequent chapters contain a more detailed description.

| Parameter | Description | |
|---|---|---|
| Variable type (Page 98) | The variable type must be specified. | |
| | R[x]: | REAL (+ digit for the decimal place) |
| | I: | INTEGER |
| | S[x]: | STRING (+ digit for string length) |
| | C: | CHARACTER (individual character) |
| | B: | BOOL |
| | V: | VARIANT |
| Limits (Page 85) | Limit value MIN, limit value MAX<br>Default setting: Empty<br>The limit values are separated by a comma. Limits can be specified for types I, C and R in decimal formats or as characters in the form "A", "F". | |
| | Optionally, two signal colors SC1 and SC2 can be configured for the display of a progress bar with color change (variable attribute DT = 1) which are displayed as the respective foreground color of the bar when the signal value SVAL1 or SVAL2 is exceeded. The signal values are specified as integer values.<br>See application example for Progress bar (Page 89). | |
| Default setting (Page 103) | If no default setting has been configured and no system or user variable has been assigned to the variable, the first element of the toggle field is assigned. If no toggle field has been defined, there is no default setting, which means the status of the variable is "not calculated".<br>Default setting: No default | |
| Toggle field (Page 101) | List with predetermined entries in the IO field: The list is initiated by a *; the entries are separated by a comma. The entries can be assigned a value.<br>For the toggle field, the entry for the limit is interpreted as a list. If only one * is entered, a variable toggle field is created.<br>Default setting: None | |
| Texts (Page 84) | The sequence is specified. Instead of a short text, an image can also be displayed.<br>Default setting: Empty | |
| | Long text: | Text in the display line |
| | Short text: | Name of the dialog element |
| | Graphic text: | Text refers to the terms in the graphics |
| | Unit text: | Unit of the dialog element |
| | Tooltip | Serves as brief information in a screen form configuration for the display and toggle fields. The information is configured via plain text and language text ID. |

<table>
<tr><th>Parameter</th><th colspan="2">Description</th></tr>
<tr><td>Attributes (Page 85)</td><td colspan="2">The attributes influence the following properties:<br>• Display mode<br>• Display option<br>• Update rate<br>• Toggle symbol<br>• Tooltip<br>• Input mode<br>• Access level<br>• Alignment of short text<br>• Font size<br>• Limits<br>The attributes are separated by commas and appear in any order. A definition can be made for each component.</td></tr>
<tr><td></td><td>Display mode</td><td>dt0: Standard (input/output field or toggle field) (default)<br>dt1: Progress bar with color change<br>dt2: Progress bar without color change in the Operate style<br>dt4: List field<br>dt5: Password input mode (asterisk)</td></tr>
<tr><td></td><td>Display option</td><td>Particularly, e.g. for the display modes dt1 and dt2 (progress bars), a display option may also have to be configured in combination.<br>do0: From left to right (default)<br>do1: From right to left<br>do2: From bottom to top<br>do3: From top to bottom</td></tr>
<tr><td></td><td>Update rate</td><td>The attribute UR (update rate) controls the update of the display and therefore the processing of the associated, configured CHANGE block of variables or grid columns. Depending on the configuration, the CPU load can be drastically reduced and therefore shorter response times achieved on the user interface.<br>ur0: SlCap::standardUpdateRate() (currently = 200 ms, default value)<br>ur1: 50 ms<br>ur2: 100 ms<br>ur3: 200 ms<br>ur4: 500 ms<br>ur5: 1000 ms<br>ur6: 2000 ms<br>ur7: 5000 ms<br>ur8: 10000 ms</td></tr>
<tr><td></td><td>Toggle symbol</td><td>tg0: Toggle symbol off (default)<br>tg1: Toggle symbol on<br>If the attribute TG is set to 1, the toggle symbol also appears in the tooltip of the input field.<br>Example:<br><code>DEF OFFS = (R//123.456/,,,,"My ToolTip"/TG1)</code></td></tr>
</table>

| Parameter | Description | |
|---|---|---|
| | Tooltip | The tooltip text can also be changed during runtime via the variable property "TT".<br>Example:<br>`PRESS(VS1)`<br>`  MyVar.TT = "My new ToolTip"`<br>`END_PRESS`<br>or<br>`DEF MyVar=(R///,,,,"My ToolTip-text")` |
| | Input mode | wr0: IO field invisible, short text visible<br>wr1: Read (no focus possible for input)<br>wr2: Read and write (line appears in white)<br>wr3: wr1 with focus<br>wr4: All variable elements invisible, no focus possible<br>wr5: The value entered is saved immediately on every keystroke (in contrast to wr2, where it is only saved when the field is exited or RETURN is pressed).<br>Default setting: wr2 |
| | Access level (Page 308) | Empty: Can always be written<br>ac0...ac7: Protection levels<br>If the access level is not adequate, then the first line is displayed in gray, default setting: ac7 |
| | Alignment of short text | al0: Left-justified<br>al1: Right-justified<br>al2: Centered<br>Default setting: al0 |
| | Font size | fs1: Default font size (8 pt.)<br>fs2: Double font size<br>Default setting: fs1<br>The clearances between the lines is defined. With the default font size, 16 lines will fit into the dialog. Graphics and unit text can only be configured in the default font size. |
| | Limits | Consequently, it is possible to check whether the values of the variable are within the MIN and MAX limits specified.<br>Default setting: Determined by specified limits<br>li0: No check<br>li1: Check with respect to min.<br>li2: Check with respect to max.<br>li3: Check with respect to min. and max. |
| | Behavior when opening | cb attributes specified for a variable in a variables definition take priority over the cb default setting in the dialog definition. Multiple attributes are separated by commas (see also Behavior when opening the dialog (attribute CB) (Page 312)). |
| | Calling the CHANGE method | CB0: The CHANGE method is triggered when the screen is displayed if the variable has a valid value at this time (e.g. through default setting or NC/PLC variable). |

| Parameter | Description | |
|---|---|---|
| | | CB1: The CHANGE method is not explicitly triggered when the screen is displayed. If the variable has a configured NC/PLC variable, then the CHANGE method is of course still called. |
| | Empty Input (EI) | The variable attribute "EI" (= Empty Input) can be used to specify how the input field is to respond if an empty string is entered.<br>EI0: Standard, i.e. empty inputs into the input field are accepted.<br>EI1: Empty inputs lead to an error status for the input field, and the previous value is set again (Undo). |
| Help display (Page 84) | Help display file: | Name of the png file<br>Default setting: Empty |
| | The name of the Help display file appears in double quotation marks. The display appears automatically (instead of the previous graphic) if the cursor is positioned on this variable. | |
| System or user variable (Page 86) | System or user data from the NC/PLC can be assigned to the variable. The system or user variable appears in double quotation marks.<br>Reference: List Manual System Variables, /PGAsl/ | |
| Position of short text (Page 104) | Position of short text (distance from left, distance from top, width)<br>The positions are entered in pixels and relate to the upper left-hand corner of the main body of the dialog. The entries are separated by commas. | |
| Position of input/output field (Page 104) | Position of input/output field (distance from left, distance from top, width, height)<br>The positions are entered in pixels and relate to the upper left-hand corner of the main body of the dialog. The entries are separated by commas. If this position changes, the positions of the short text, graphic text and unit text also change. | |
| Colors (Page 84) | Foreground and background colors for input/output fields, short texts, graphic texts, unit texts and signal colors for progress bars:<br>The colors are separated by a comma.<br>For specification of the color, see Chapter List of colors (Page 309).<br>Default setting for input/output field: Foreground color: Black, background color: White<br>The default colors of the input/output field depend on the "wr" write mode:<br>Default setting for short text, graphic text, unit text: Foreground color: Black, background color: Transparent. | |
| | Optionally, two signal colors SC1 and SC2 can be configured for the display of a progress bar with color change (variable attribute DT = 1) which are displayed as the respective foreground color of the bar when the signal value SVAL1 or SVAL2 is exceeded.<br>See application example for Progress bar (Page 89). | |
| | The colors are expected in the following order in the variables definition:<br>1. Foreground color of the input/output field: FC<br>2. Background color of the input/output field: BC<br>3. Foreground color of the short text: FC_ST<br>4. Background color of the short text: BC_ST<br>5. Foreground color of the graphic text: FC_GT<br>6. Background color of the graphic text: BC_GT<br>7. Foreground color of the unit text: FC_UT<br>8. Background color of the unit text: BC_UT<br>9. Signal color 1<br>10. Signal color 2 | |

| Parameter | Description |
|---|---|
| Online help file | The name of the online help file appears in double quotation marks. |
| I/O field with integrated unit selection | You can use the input/output fields with integrated unit selection to toggle between different units. If the cursor is moved to the input field, the integrated unit selection is highlighted (focus does not have to be explicitly placed on it). A tooltip with a toggle symbol is also displayed with the appropriate reference to this functionality.<br>Examples:<br>`DEF VarEdit=(R/////////200,,100///"VarTgl")`<br>`VarTgl=(S/*0="mm",1="inch"/0//WR2////302,,40)`<br>or<br>`DEF VarEdit_2={TYP="R", VAL=1.234, X=200, W=100, LINK_TGL="vartgl_2"},`<br>`VARTgl_2={TYP="S", TGL="* 0=""mm"", 1=""inch""",WR=2, X=302, W=40}` |

## Variable: Changing properties

The variables are assigned a new value in the notation *Identifier.Property = Value* when a change is made. The expression to the right of the equality sign is evaluated and assigned to the variable or variable property.

| | |
|---|---|
| Identifier.**ac** = Access level | (ac: access level) |
| Identifier.**al** = Text alignment | (al: alignment) |
| Identifier.**bc** = Background color of the input/output field | (bc: back color) |
| Identifier.**bc_gt** = Background color of the graphic text | (bc: back color)<br>(gt: graphic text) |
| Identifier.**bc_st** = Background color of the short text | (bc: back color)<br>(st: short text) |
| Identifier.**bc_ut** = Background color of the unit text | (bc: back color)<br>(ut: unit text) |
| Identifier.**do** = Display option | (do: display option) |
| Identifier.**dt** = Display mode | (dt: display type) |
| Identifier.**fc** = Foreground color of the input/output field | (fc: front color) |
| Identifier.**fc_gt** = Foreground color of the graphic text | (fc: front color)<br>(gt: graphic text) |
| Identifier.**fc_st** = Foreground color of the short text | (fc: front color)<br>(st: short text) |
| Identifier.**fc_ut** = Foreground color of the unit text | (fc: front color)<br>(ut: unit text) |
| Identifier.**fs** = Font size | (fs: font size) |
| Identifier.**gt** = Graphic text | (gt: graphic text) |
| Identifier.**hlp** = Help display | (hlp: help) |
| Identifier.**li** = Limit | (li: limit) |
| Identifier.**lt** = Long text | (lt: long text) |
| Identifier.**max** = MAX limits | (max: maximum) |
| Identifier.**min** = MIN limits | (min: minimum) |

| | |
|---|---|
| Identifier.**sc** = Signal color | (sc: signal color) |
| Identifier.**st** = Short text | (st: short text) |
| Identifier.**tg** = Toggle symbol | (tg: toggle) |
| Identifier.**tt** = Tooltip | (tt: tooltip |
| Identifier .**typ** = Variable type | (typ: type) |
| Identifier.**ur** = Update rate | (ur: update rate) |
| Identifier.**ut** = Unit text | (ut: unit text) |
| Identifier.**val** = Variable value | (val: value) |
| Identifier.**var** = System or user variable | (var: variable) |
| Identifier.**vld** = Variable status | (vld: validation) |
| Identifier.**wr** = Input mode | (wr: write) |

## See also

Extended configuration syntax (Page 43)

# 5.11 Details on the variable type

## Variable type INTEGER

The following extensions for determining the display in the input/output field and the memory utilization are possible for the "INTEGER" type:

**2nd character** in the extension data type

| Display format | |
|---|---|
| B | Binary |
| D | Decimal signed |
| H | Hexadecimal |
| No data | Decimal signed |

**3rd and/or 4th character** in the extension data type

| Memory utilization | |
|---|---|
| B | Byte |
| W | Word |
| D | Double word |
| BU | Byte, Unsigned |
| WU | Word, Unsigned |
| DU | Double word, Unsigned |

## Sequence of characters in the INTEGER data type

1. "I" Basic INTEGER designation
2. Display format
3. Memory utilization
4. "U" Unsigned

| Valid INTEGER type specifications: | |
|---|---|
| IB | Integer variable 32 bits in binary notation |
| IBD | Integer variable 32 bits in binary notation |
| IBW | Integer variable 16 bits in binary notation |
| IBB | Integer variable 8 bits in binary notation |
| I | Integer variable 32 bits in decimal notation signed |
| IDD | Integer variable 32 bits in decimal notation signed |
| IDW | Integer variable 16 bits in decimal notation signed |
| IDB | Integer variable 8 bits in decimal notation signed |
| IDDU | Integer variable 32 bits in decimal notation unsigned |
| IDWU | Integer variable 16 bits in decimal notation unsigned |
| IDBU | Integer variable 8 bits in decimal notation unsigned |

| Valid INTEGER type specifications: | |
|---|---|
| IH | Integer variable 32 bits in hexadecimal notation |
| IHDU | Integer variable 32 bits in hexadecimal notation |
| IHWU | Integer variable 16 bits in hexadecimal notation |
| IHBU | Integer variable 8 bits in hexadecimal notation |

## VARIANT variable type

The VARIANT variable type is determined by the data type of the last value assignment. If the assigned or entered value starts with '-', '+', '.' or a number ('0'-'9'), then the value is interpreted as numeric. In all other cases as a string.

It can be scanned using the ISNUM or ISSTR functions. The VARIANT type is mainly suited to the purpose of writing either variable names or numerical values to the NC code.

## Programming

The data type of variables can be checked:

| Syntax: | **ISNUM** *(VAR)* | |
|---|---|---|
| Parameter: | VAR | Name of the variable whose data type is to be checked. |
| | | The result of the scan can be: |
| | FALSE = | not a numerical variable (data type = STRING) |
| | TRUE = | numerical variable (data type = REAL) |

| Syntax: | **ISSTR** *(VAR)* | |
|---|---|---|
| Parameter: | VAR | Name of the variable whose data type is to be checked. |
| | | The result of the scan can be: |
| | FALSE = | numerical variable (data type = REAL) |
| | TRUE = | not a numerical variable (data type = STRING) |
| Example: | | |

```
IF ISNUM(VAR1) == TRUE
IF ISSTR(REG[4]+2) == TRUE
```

The display mode of variables can be changed:

- For INTEGER, the display type can be changed.

| | |
|---|---|
| B | Binary |
| D | Decimal signed |
| H | Hexadecimal |
| unsigned | |
| With the addition of U for Unsigned | |

- For REAL data types, only the number of places after the decimal point can be changed. Changing the type is illegal and generates an error message in the "easyscreen_log.txt" file.
  Example:
  ```
  Var1.typ = "IBW"
  Var2.typ = "R3"
  ```

## Numerical formats

Numbers can be represented in either binary, decimal, hexadecimal or exponential notation:

| | |
|---|---|
| Binary | B01110110 |
| Decimal | 123.45 |
| Hexadecimal | HF1A9 |
| exponential | -1.23EX-3 |
| Examples: | |
| | `VAR1 = HF1A9` |
| | `REG[0]= B01110110` |
| | `DEF VAR7 = (R//-1.23EX-3)` |

**Note**

When codes are generated with the "GC" function, only numerical values in decimal or exponential notation are evaluated, but **not** those in binary or hexadecimal notation.

## See also

Variable parameters (Page 92)

# 5.12 Details on the toggle field

## Description

The toggle field extension function displays texts (entries in toggle field) as a function of NC/PLC variables. A variable, which makes use of a toggle field extension, is read-only. The list of the toggle field is opened by pressing the ENTER key.

## Programming

| | | |
|---|---|---|
| Syntax: | **DEF VAR1=(IB/+ $85000/15////"DB90.DBB5")** or<br>**DEF VAR_TGL = (S/* "Hello", "Run", "MyScreens"/"Run")** | |
| Description: | When the dialog is opened, the content of text number $85015 is displayed in the IO field. Default value 15 is entered in system variable DB90.DBB5. If the value saved in system variable DB90.DBB5 changes, the displayed text number $(85000 + <DB90.DBB5>) is recalculated in response to every change. | |
| Parameter: | Variable type | Type of variables specified in the system or user variable. |
| | Text number | Number (basis) of the language-specific text valid as the basis number. |
| | System or user variable | System or user variable (offset) via which the final text number (basis + offset) is displayed. |

## Toggle-field-dependent displays

The toggle field is overlaid with graphics, which change depending on the value of the memory byte. If the value of the memory byte is 1, "image1.png" will appear. If it is 2, "image2.png" will appear.

```
DEF VAR1=(IDB/*1="\\image1.png", 2="\\image2.png"//,$85000/wr1//"MB[130]"//160,40,50,50)
```

The size and position of the image is defined under "Position of IO field (left, top, width, height)".

## Virtual toggle key

There is no list for toggle fields without configured list, e.g. `DEF NoTglList=(R/*)`. The next element is not generated until the toggle key is actuated or the associated CHANGE() method of the corresponding variable run through.

For this case, a small virtual keyboard (comprising only a toggle key) is displayed on the right next to the variable toggle field for operation with a touch panel.

The display of the virtual toggle key can also be forced via the following entry in the configuration file "slguiconfig.ini" irrespective of a touch panel.

```
[VirtualKeyboard]
```

```
; Forces easyscreen virtual toggle keyboard function
ForceEasyscreenVirtualToggleKey = true
```

## See also

Variable parameters (Page 92)

# 5.13 Details on the default setting

## Overview

A variable can assume various states depending on whether a default value, or a system or user variable, or both, is assigned to the variable field (input/output field or toggle field), (not calculated: toggling is only possible if a valid value has been assigned to the variable).

## Scope of the default settings

| Condition | | | Reaction of field type |
|---|---|---|---|
| Field type | Default setting | System or user variable | |
| I/O field | Yes | Yes | Write default value to system or user variable |
| | No | Yes | Use system or user variable as default value |
| | Fault | Yes | Not calculated, system or user variable is not written/ used. |
| | Yes | No | Default setting |
| | No | No | Not calculated |
| | Fault | No | Not calculated |
| | Yes | Fault | Not calculated |
| | No | Fault | Not calculated |
| | Fault | Fault | Not calculated |
| Toggle | Yes | Yes | Write default value to system or user variable |
| | No | Yes | Use system or user variable as default value |
| | Fault | Yes | Not calculated,<br>system or user variable not written/used |
| | Yes | No | Default setting |
| | No | No | Default = first toggle field element |
| | Fault | No | Not calculated |
| | Yes | Fault | Not calculated |
| | No | Fault | Not calculated |
| | Fault | Fault | Not calculated |

## See also

Variable parameters (Page 92)

# 5.14 Details on the position of the short text, position of the input/output field

## Overview

The short text and graphic text, as well as the input/output field and unit text, are each treated like a unit, i.e., position settings for short text apply to the graphic text and data for the input/output field and to unit text.

## Programming

The configured position entry overwrites the default value, i.e., only one value can be changed. If no position settings have been configured for subsequent screen form elements, then the position settings for the preceding screen form element are applied.

The default setting is used if positions are not specified for any of the dialog elements. The column width for the short text and input-output field is calculated as standard for each line from the number of columns and maximum line width, i.e. column width = maximum line width/number of columns.

The width of the graphics and unit text is predefined and optimized to suit the requirements of programming support. If graphics or unit text has been configured, the width of the short text or I/O field is reduced accordingly.

The order of short text and I/O field can be reversed by position settings.

## Distance between the input/output field and unit field and width of the unit field

You can configure the distance between an input/output field and a unit field as well as the width of the unit field.

In the definition line, in the section for the input/output position, enter the distance from the input/output field to the unit field, separated by a comma (e.g. 7 pixels) and/or the width of the unit field (e.g. 60 pixels):

```
DEF VarDT=(R3//0.000/,"DT",,"s"////0,,24/39,,71,,7,60)
```

Or:

```
DEF VarDT={TYP="R3", VAL="0.000", ST="DT", UT="s", TXT_X=0,
TXT_W=24, X=39, W=71, UT_DX=7, UT_W=60}
```

In this case, the distance between the input/output field/unit field and/or the width of the unit field have/has been configured and the following points should be taken into account:

- The configured width of the input/output field does **not** include the fixed width of the unit field (this is fixed at 50 pixels). This means that you directly configure the width of the input/output field.
- A 50 pixel width applies as default if a width is not configured for the unit field.
- 0 pixels applies as default if a distance is not configured between the input/output field / unit field.

**Special feature regarding the associated toggle field**

Requirements:

- An associated toggle field is configured for a variable,
- For the variable, a distance is configured between the input/output field and unit field and/or a width for the unit field, and
- The variable has no unit text.

In this particular case, the associated toggle field is positioned at the configured position of the unit field of the variable.

A position possibly configured for the input/output component of the associated toggle field is ignored.

## Example

In the following example, the associated toggle field **F_Unit** is automatically set with a distance of **7 pixels** to the input/output field of variable **VarF** , and with a width of **59 pixels** .

```
DEF VarF=(R//0.0/,"F",,,////0,,24/39,,85,,7,59///"F_Unit"), F_Unit =
(I/*3="mm/min", 1="mm/U"/3// ////181,,155)
```

## See also

Variable parameters (Page 92)

# 5.15 Use of strings

## Strings

Strings can be used as part of the configuration. These allow text to be displayed dynamically or different texts to be chained for the purpose of code generation.

## Rules

The following rules must be observed with regard to string variables:

- Logic operations are processed from left to right.
- Nested expressions are solved from the inside outwards.
- No distinction is made between uppercase and lowercase type.
- String variables are generally displayed left justified.

Strings can be deleted simply by assigning a blank string.

Strings can be appended after the equality sign using the operator "<<". Quotation marks (") in the string are represented by two successive quotation mark symbols. Strings can be checked for equality in IF instructions.

## Example

Default settings for the following examples:

```
VAR1.VAL = "This is an"
VAR8.VAL = 4
VAR14.VAL = 15
VAR2.VAL = "Error"
$85001 = "This is an"
$85002 = "Alarm text"
```

Editing strings:

- Chaining of strings:
  ```
  VAR12.VAL = VAR1 << " Error."   ;Result: "This is an error"
  ```
- Deleting a variable:
  ```
  VAR10.VAL = ""   ;Result: Blank string
  ```
- Setting a variable with a text variable:
  ```
  VAR11.VAL = VAR1.VAL   ;Result: "This is an"
  ```
- Data type matching:
  ```
  VAR13.VAL ="This is the " << (VAR14 - VAR8) << ". error"
              ;Result: "This is the 11th error"
  ```

- Treatment of numerical values:

```
VAR13.VAL = "Error" << VAR14.VAL << ": " << $85001 << $85002
         ;Result: "Error 15: "This is an alarm text"
IF VAR15 == "Error"   ;Strings in IF statement
  VAR16 = 18.1234
         ;Result: VAR16 equals 18.1234,
         ;if VAR15 equals "Error".
ENDIF
```

- Quotation marks within a string:

```
VAR2="Hello, this is a " Test""
   ;Result: Hello, this is a " Test"
```

- System or user-variable strings dependent on variable content:

```
VAR2.Var = "$R[" << VAR8 << "]"   ;Result: $R[4]
```

## See also

STRING functions (Page 182)

# 5.16 CURPOS variable

## Description

The CURPOS variable calls or manipulates the position of the cursor in the active input field of the current dialog. The variable shows how many characters are in front of the cursor. If the cursor is located at the start of the input field, then CURPOS assumes the value of 0. If the value of CURPOS is changed, then the cursor is positioned at the appropriate location in the input field.

In order to be able to respond to changes in the variable value, it is possible to monitor for changes using a CHANGE method. If the value of CURPOS changes, then a jump is made to the CHANGE method and the instructions contained there are executed.

# 5.17 CURVER variable

## Description

The CURVER (CURrent VERsion) property allows the programming to be adapted in order to handle different versions. The CURVER variable is read-only.

**Note**

Even if previously recompiled with an older version, the code is automatically generated with the most recent version. The "GC" command always generates the most recent version. An additional identifier indicating the generated version is inserted in the user comment of the generated code in versions > 0.

## Rules

The most recent dialog with all its variables is always displayed.

- Variables used previously may not be changed.
- New variables are inserted in the existing (cycle) programming in arbitrary order.
- It is not permissible to delete variables from a dialog from one version to the next.
- The dialog must contain all variables of all versions.

## Example

```
(IF CURVER==1 ...)                ; When the code is recompiled, CURVER is automat-
                                    ically assigned the version of the recompiled
                                    code.
```

# 5.18 ENTRY variable

## Description

The ENTRY variable can be used to check by what method a dialog has been called.

## Programming

| Syntax: | **ENTRY** | |
|---|---|---|
| Description: | The ENTRY variable is a read only variable. | |
| Return Value: | | The result of the scan can be: |
| | 0 = | No programming support |
| | 1 = | Start of a screen via softkey; no code has been generated yet (default as configured) |
| | 2 = | Start of a screen via softkey; code has been generated<br>(default from the code generated last by this screen) |
| | 3 = | Recompilation with user comment (# lines) |
| | 4 = | Code_type = 0: NC code with user comment (# lines) |
| | 5 = | Code_type = 1: NC code without user comment (# lines) |

## Example

```
IF ENTRY == 0
  DLGL ("The dialog was not called under programming")
ELSE
  DLGL ("The dialog was called under programming")
ENDIF
```

# 5.19 ERR variable

## Description

Variable ERR checks whether the preceding line has been executed correctly.

## Programming

| | | |
|---|---|---|
| Syntax: | **ERR** | |
| Description: | The ERR variable is read-only. | |
| Return value: | | The result of the scan can be: |
| | FALSE = | previous line was executed error-free |
| | TRUE = | previous line was not executed error-free |

## Example

```
VAR4 = Thread[VAR1,"CDM",3]            ; Output value from array
IF ERR == TRUE                         ; Query whether the value has been found in
                                       the array
  VAR5 = "Error accessing array"
                                       ; If the value has not been found in the ar-
                                       ray, the value "Error accessing array" is as-
                                       signed to the variables.
ELSE
  VAR5 = "All OK"                      ; If the value has been found in the array,
                                       the value "All OK" is assigned to the varia-
                                       bles.
ENDIF
```

# 5.20 FILE_ERR variable

## Description

Variable FILE_ERR can be used to check whether the preceding GC or CP command has been executed correctly.

## Programming

| Syntax: | **FILE_ERR** | |
|---|---|---|
| Description: | The FILE_ERR variable is read-only. | |
| Return value: | | Possible results are: |
| | 0 = | Operation okay |
| | 1 = | Drive/path not available |
| | 2 = | Path/file access error |
| | 3 = | Drive not ready |
| | 4 = | Incorrect file name |
| | 5 = | File is already open |
| | 6 = | Access denied |
| | 7 = | Target path not available or not permitted |
| | 8 = | Copy source same as target |
| | 10 = | Internal error: FILE_ERR = 10 means that the error cannot be classified in the other categories. |

## Example

```
CP("D:\source.mpf","E:\target.mpf")                    ; Copy from source.mpf to E:\target.mpf
IF FILE_ERR > 0                                        ; Query whether error has occurred
   IF FILE_ERR == 1                                    ; Query specific error numbers and output associated error text
      VAR5 = "Drive/path not available"
   ELSE
      IF FILE_ERR == 2
         VAR5 = "Path/file access error"
      ELSE
         IF FILE_ERR == 3
            VAR5 = "incorrect file name"
         ENDIF
      ENDIF
   ENDIF
ELSE
```

```
CP("D:\source.mpf","E:\target.mpf")
   VAR5 = "All OK"                          ; If no errors have occurred in CP
                                            (or GC), "All OK" is output
ENDIF
```

# 5.21 FOC variable

## Description

With the variable FOC, the input focus (current active input/output field) is controlled in a dialog. The reaction of the cursor left, right, up, down as well as PGUP, PGDN are predefined.

**Note**

The FOC function must not be initiated as a result of a navigation event. The cursor position may only be changed in softkey PRESS methods, CHANGE methods, etc.

The FOC function cannot be applied to variables with input mode wr = 0 and wr = 4 or to Help variables.

## Programming

| Syntax: | **FOC** | |
|---|---|---|
| Description: | The variable can be read and written. | |
| Return value: | Read | The result is the name of the variable to which the FOC function has been applied. |
| | Write | It is possible to assign either a string or a numerical value. A string is interpreted as a variable name and a numerical value as a variable index. |

## Example

```
IF FOC == "Var1"              ; Read focus
  REG[1] = Var1
ELSE
  REG[1] = Var2
ENDIF

FOC = "Var1"                  ; The input focus is assigned to variable 1.
FOC = 3                       ; The input focus is assigned to the 3rd dialog ele-
                              ment with WR ≥ 2.
```

## See also

FOCUS (Page 129)

# 5.22 Variable S_ALEVEL

## Description

The current access level can be queried in the configuration with the screen property S_ALEVEL.

## Programming

| | |
|---|---|
| Syntax: | **S_ALEVEL** |
| Description: | Query of the current access level |
| Return value: | 0: System<br>1: Manufacturer<br>2: Service<br>3: User<br>4: Key switch 3<br>5: Key switch 2<br>6: Key switch 1<br>7: Key switch 0 |

## Example

```
REG[0] = S_ALEVEL
```

## See also

ACCESSLEVEL (Page 126)

# 5.23 S_CHAN variable

## Description

The S_CHAN variable determines the number of the current channel for display or evaluation purposes.

## Programming

| | |
|---|---|
| Syntax: | **S_CHAN** |
| Description: | Query of the current channel number |
| Return value: | Channel number |

## Example

```
REG[0] = S_CHAN
```

## See also

CHANNEL (Page 128)

# 5.24 Variable S_CONTROL

## Description

The current control name can be queried in the configuration with the screen property S_CONTROL.

## Programming

| | |
|---|---|
| Syntax: | **S_CONTROL** |
| Description: | Query of the current control name |
| Return value: | The control name is the section name in the "mmc.ini" |

## Example

```
REG[0] = S_CONTROL
```

## See also

CONTROL (Page 129)

# 5.25 Variable S_LANG

## Description

The current language can be queried in the configuration with the screen property S_LANG.

## Programming

| | |
|---|---|
| Syntax: | **S_LANG** |
| Description: | Query of the current language |
| Return value: | Language code from resources.xml, e.g. "deu", "eng", etc. |

## Example

```
REG[0] = S_LANG
```

## See also

LANGUAGE (Page 130)

# 5.26 Variable S_NCCODEREADONLY

## Description

The screen property S_NCCODEREADONLY is only relevant when a cycle is recompiled in the editor with a "Run MyScreens" dialog. S_NCCODEREADONLY determines whether a recompiled NC code from the editor can be changed or not.

The following applies for the return value:

- TRUE: The recompiled NC code from the editor can be changed
- FALSE: The recompiled NC code from the editor cannot be changed (read-only), because, for example, it is already in the preprocessing (= TRUE).

# 5.27 Variables S_RESX and S_RESY

## Description

The current resolution or its X and Y component can be queried in the configuration with the screen properties S_RESX and S_RESY.

## Example

```
REG[0] = S_RESY
```

## See also

RESOLUTION (Page 135)

# Programming commands

# 6

## 6.1 Operators

### Overview

The following operators can be used when programming:

- Mathematical operators
- Relational operators
- Logic (Boolean) operators
- Bit operators
- Trigonometric functions

## 6.1.1 Mathematical operators

### Overview

| Mathematical operators | Identifier |
|---|---|
| + | Addition |
| - | Subtraction |
| * | Multiplication |
| / | Division |
| MOD | Modulo operation |
| ( ) | Parentheses |
| AND | AND operator |
| OR | OR operator |
| NOT | NOT operator |
| ROUND | Round off numbers with decimal places |

**Example:** `VAR1.VAL = 45 * (4 + 3)`

### ROUND

The ROUND operator is used to round off numbers with up to 12 decimal places during execution of a dialog configuration. The variable fields cannot accept the decimal places in the display.

## Use

ROUND is controlled by the user with two parameters:

```
VAR1 = 5.2328543
VAR2 = ROUND( VAR1, 4 )
```

**Result:** `VAR2 = 5.2339`

VAR1 contains the number to be rounded. The parameter “4” indicates the number of decimal places in the result, which is placed in VAR2.

## Trigonometric functions

| Trigonometric functions | Identifier |
|---|---|
| SIN(x) | Sine of x |
| COS(x) | Cosine of x |
| TAN(x) | Tangent of x |
| ATAN(x, y) | Arc tangent of x/y |
| SQRT(x) | Square root of x |
| ABS(x) | Absolute value of x |
| SDEG(x) | Conversion to degrees |
| SRAD(x) | Conversion to radian |
| CALC_ASIN(x) | Arc sine of x |
| CALC_ACOS(x) | Arc cosine of x |

**Note**

The functions operate with radian measure. The functions SDEG() and SRAD() can be used for conversion.

**Example:** `VAR1.VAL = SQRT(2)`

## Mathematical functions

| Mathematical functions | Identifier |
|---|---|
| CALC_CEIL(x) | Determines the next highest integer of x (round up) |
| CALC_FLOOR(x) | Determines the next lowest integer of x (round down) |
| CALC_LOG(x) | Determines the (natural) logarithm for the base e of x |
| CALC_LOG10(x) | Determines the logarithm for the base 10 of x |
| CALC_POW(x, y) | Determines x to the power of y |
| CALC_MIN(x, y) | Determines the lower number of x and y |
| CALC_MAX(x, y) | Determines the higher number of x and y |

## Random function: Random number

| Syntax: | **RANDOM** *(lower limit value, upper limit value)* | |
|---|---|---|
| Description: | The **RANDOM** function returns a pseudo random number in a specified range. | |
| Parameters: | Lower limit value | Lower limit value >= -32767,<br>Lower limit value < upper limit value |
| | Upper limit value | Upper limit value <= 32767 |

## Example

```
REG[0] = RANDOM(-10,10)         ; Possible result = -3
```

## *Constants*

| *Constants* | |
|---|---|
| *PI* | *3.14159265358979323846* |
| *FALSE* | *0* |
| *TRUE* | *1* |

*Example:* `VAR1.VAL = PI`

## *Comparison operators*

| *Comparison operators* | |
|---|---|
| *==* | *Equal to* |
| *<>* | *Not equal to* |
| *>* | *Greater than* |
| *<* | *Less than* |
| *>=* | *Greater than or equal to* |
| *<=* | *Less than or equal to* |

*Example:*

```
IF VAR1.VAL == 1
  VAR2.VAL = TRUE
ENDIF
```

### *Conditions*

*The nesting depth is unlimited.*

*Condition with one command:*

```
IF
...
ENDIF
```

*Condition with two commands:*

```
IF
...
ELSE
...
ENDIF
```

## 6.1.2 Bit operators

### Overview

| Bit operators | Identifier |
|---|---|
| BOR | Bit-serial OR |
| BXOR | Bit-serial XOR |
| BAND | Bit-serial AND |
| BNOT | Bit-serial NOT |
| SHL | Shift bits to left |
| SHR | Shift bits to right |

### SHL operator

Bits are shifted to the left using the SHL (SHIFT LEFT) operator. You can specify both the value to be shifted and the number of shift increments directly or via a variable. If the limit of the data format is reached, the bits are shifted beyond the limit without displaying an error message.

### Use

| | | |
|---|---|---|
| Syntax: | variable = *value* **SHL***increment* | |
| Description: | Shift Left | |
| Parameters: | value | value to be shifted |
| | increment | number of shift increments |

### Example

```
PRESS(VS1)
```

```
  VAR01 = 16 SHL 2           ; Result = 64
  VAR02 = VAR02 SHL VAR04    ; Convert content of VAR02 to 32-bit unsigned
                             , and shift content to left by number of bits specified
                             in VAR04. Then convert 32-bit value back to format of
                             variable VAR02.
END_PRESS
```

## SHR operator

Bits are shifted to the RIGHT using the SHR (SHIFT RIGHT) function. You can specify both the value to be shifted and the number of shift increments directly or via a variable. If the limit of the data format is reached, the bits are shifted beyond the limit without displaying an error message.

## Use

| | | | |
|---|---|---|---|
| Syntax: | variable = *value* **SHR***increment* | | |
| Description: | Shift Right | | |
| Parameters: | value | value to be shifted | |
| | increment | number of shift increments | |

## Example

```
PRESS(VS1)
  VAR01 = 16 SHR 2               ; Result = 4
  VAR02 = VAR02 SHR VAR04        ; Convert content of VAR02 to 32-bit unsigned
                                 , and shift content to right by number of bits
                                 specified in VAR04. Then convert 32-bit value back
                                 to format of variable VAR02.
END_PRESS
```

# 6.2 Methods

## Overview

Various types of event (exit input field, actuate softkey) can initiate specific actions in dialogs and dialog-dependent softkey menus (softkey menus that are called from a newly configured dialog). These actions are configured in methods.

The following table shows the basic principle used to program a method:

| Definition block | Comment | Section reference |
|---|---|---|
| `PRESS(HS1)` | ;Method start identifier | |
| `LM...`<br>`  LS...` | ;Functions | See Chapter Functions (Page 138) |
| `Var1.st = ...` | ;Changing properties | see Section Defining softkey menus (Page 62)<br>and Section Defining dialog properties (Page 49). |
| `Var2 = Var3 + Var4`<br>`...`<br>`EXIT` | ;Calculation with variables | See Chapter Defining variables (Page 81) |
| `END_PRESS` | ;Method end identifier | |

## 6.2.1 ACCESSLEVEL

### Description

The ACCESSLEVEL method is run through when the current access level has changed for an opened screen.

### Programming

| | |
|---|---|
| Syntax: | ACCESSLEVEL<br><instructions><br>END_ACCESSLEVEL |
| Description: | Access level |
| Parameters: | - None - |

### See also

Variable S_ALEVEL (Page 115)

## 6.2.2 CHANGE

### Description

CHANGE methods are executed if a variable value changes, i.e., variable calculations that are performed as soon as a variable value changes are configured within a CHANGE method.

There are two types of CHANGE method, i.e., element-specific and global:

- The **element-specific CHANGE method** is executed if the value of a specified variable changes. If a system or user variable is assigned to a variable, cyclic updating of the variable value can be configured in a CHANGE method.
- The **global CHANGE method** is executed if the value of any variable changes and no element-specific CHANGE method has been configured.

### "Element-specific" programming

| | | |
|---|---|---|
| Syntax: | CHANGE*(Identifier)*<br>...<br>END_CHANGE | |
| Description: | Changes the value of a specific variable | |
| Parameters: | Identifier | Name of the variable |

### Example

```
DEF VAR1=(I//////"DB20.DBB1")        ; A system variable is assigned to Var1
CHANGE(VAR1)
  IF VAR1.Val <> 1
  VAR1.st="Tool OK!"                 ; If the value of the system variable ≠ 1, the
                                     short text of the variable states: Tool OK!
    otto=1
  ELSE
    VAR1.st="Attention: Error!"      ; If the value of the system variable = 1, the
                                     short text of the variable states: Attention:
                                     Error!
    otto=2
  ENDIF
  VAR2.Var=2
END_CHANGE
```

### "Global" programming

| | |
|---|---|
| Syntax: | CHANGE()<br>...<br>END_CHANGE |
| Description: | Changes any variable value |
| Parameters: | - None - |

### Example

```
CHANGE()
  EXIT                              ; If any of the variable values change, the di-
                                    alog will be terminated.
END_CHANGE
```

### See also

Behavior when opening the dialog (attribute CB) (Page 312)

## 6.2.3 CHANNEL

### Description

The CHANNEL method is run through when the current channel has changed for an opened screen, i.e. a channel switchover has been performed.

### Programming

| | |
|---|---|
| Syntax: | CHANNEL<br><instructions><br>END_CHANNEL |
| Description: | Channel switchover |
| Parameters: | - None - |

### See also

S_CHAN variable (Page 116)

## 6.2.4 CONTROL

### Description

The CONTROL method is run through when the current control has changed for an opened screen, i.e. typically at a 1:n switchover.

### Programming

| | |
|---|---|
| Syntax: | CONTROL<br><instructions><br>END_CONTROL |
| Description: | Control switchover |
| Parameters: | - None - |

### See also

Variable S_CONTROL (Page 117)

## 6.2.5 FOCUS

### Description

The FOCUS method is executed if the focus (cursor) is positioned on another field in the dialog.

The FOCUS method must not be initiated as a result of a navigation event. The cursor position may only be changed in softkey PRESS methods, CHANGE methods, etc. The response of cursor movements is predefined.

**Note**

Within the FOCUS method, it is not possible to select a different variable, nor can a new dialog be loaded.

### Programming

| | |
|---|---|
| Syntax: | FOCUS<br>...<br>END_FOCUS |
| Description: | Positions the cursor |
| Parameters: | - None - |

### Example

```
FOCUS
  DLGL("The focus has been placed on variable << FOC << ".)
END_FOCUS
```

### See also

FOC variable (Page 114)

## 6.2.6 LANGUAGE

### Description

The LANGUAGE method is run through when the current language has changed for an opened screen.

### Programming

| | |
|---|---|
| Syntax: | LANGUAGE<br><instructions><br>END_LANGUAGE |
| Description: | Language |
| Parameters: | - None - |

### See also

Variable S_LANG (Page 118)

## 6.2.7 LOAD

### Description

The LOAD method is executed after the variable and softkey definitions (DEF Var1= ..., HS1= ...) have been interpreted. At this time, the dialog is not yet displayed.

### Programming

| | |
|---|---|
| Syntax: | LOAD<br>...<br>END_LOAD |
| Description: | Download |
| Parameters: | - None - |

### Example

```
LOAD                          ; Start identifier
  Screen form1.Hd =  $85111   ; Assign text for dialog header from language file
  VAR1.Min = 0                ; Assign MIN variable limit
  VAR1.Max = 1000             ; Assign MAX variable limit
END_LOAD                      ; End code
```

### See also

Line, dividing line, rectangle, circle and ellipse (Page 193)

## 6.2.8 UNLOAD

### Description

The UNLOAD method is executed before a dialog is unloaded.

### Programming

| | |
|---|---|
| Syntax: | UNLOAD<br>...<br>END_UNLOAD |
| Description: | Unload |
| Parameters: | - None - |

### Example

```
UNLOAD
  REG[1] = VAR1            ; Save variable in register
END_UNLOAD
```

## 6.2.9 OUTPUT

### Description

The OUTPUT method is executed if the "GC" function is called. Variables and Help variables are configured as an NC code in an OUTPUT method. The individual elements in a code line are linked by means of blanks.

**Note**

The NC code can be generated in an extra file by means of file functions and transferred to the NC.

### Programming

| | | |
|---|---|---|
| Syntax: | OUTPUT *(Identifier)*<br>...<br>END_OUTPUT | |
| Description: | Outputs variables in the NC program. | |
| Parameters: | Identifier | Name of OUTPUT method |

### Block numbers and skip identifiers

The OUTPUT method must not contain line numbers or skip identifiers if you wish to keep the line numbers and skip identifiers directly set with active program support in the parts program in case of recompilations.

Editor changes in the parts program produce the following response:

| Condition | Response |
|---|---|
| Number of blocks remains unchanged. | Block numbers are retained. |
| Number of blocks is reduced. | The highest block numbers are canceled. |
| Number of blocks is increased. | New blocks are not numbered. |

### Example

```
OUTPUT(CODE1)
  "CYCLE82(" Var1.val "," Var2.val "," Var3.val ","Var4.val "," Var5.val "," Var6.val ")"
END_OUTPUT
```

## 6.2.10 PRESS

### Description

The PRESS method is executed when the corresponding softkey is pressed.

### Programming

| | | | |
|---|---|---|---|
| Syntax: | PRESS*(softkey)*<br>...<br>END_PRESS | | |
| Designation: | Pressing a softkey | | |
| Parameter: | Softkey | Name of softkey: HS1 - HS8 and VS1 - VS8 | |
| | RECALL | <RECALL> key | |
| | ENTER | For the <ENTER> key, see PRESS(ENTER) (Page 134) | |
| | TOGGLE | For the <TOGGLE> key, see PRESS(TOGGLE) (Page 134) | |
| | PU | Page Up | Screen up |
| | PD | Page Down | Screen down |
| | SL | Scroll left | Cursor left |
| | SR | Scroll right | Cursor right |
| | SU | Scroll up | Cursor up |
| | SD | Scroll down | Cursor down |

### Example

```
HS1 = ("another softkey menu")
HS2=("no function")
PRESS(HS1)
  LS("Menu1")                    ; Load another softkey menu
  Var2 = Var3 + Var1
END_PRESS
PRESS (HS2)
END_PRESS
PRESS(PU)
  INDEX = INDEX -7
  CALL("UP1")
END_PRESS
```

## 6.2.11 PRESS(ENTER)

### Description

The PRESS(ENTER) method is always called when the Enter key is pressed for a variable with input/output field with input mode WR3 or WR5:

- WR3: Navigation to field and pressing of the Enter key
- WR5: In the input mode, transfer of the value with the Enter key

### Programming

| | |
|---|---|
| Syntax: | PRESS(ENTER)<br><instructions><br>END_PRESS |
| Description: | Enter key pressed |
| Parameters: | - None - |

## 6.2.12 PRESS(TOGGLE)

### Description

The PRESS(TOGGLE) method is always called when the toggle key is pressed irrespective of the currently focused variable.

When required, the FOC screen property can be used to determine which variable is currently in focus.

### Programming

| | |
|---|---|
| Syntax: | PRESS(TOGGLE)<br><instructions><br>END_PRESS |
| Description: | Toggle key pressed |
| Parameter: | - None - |

### Example

```
PRESS(TOGGLE)
   DLGL("Toggle key pressed at variable " << FOC)        ; The FOC screen property determines which
                                                         variable is currently in focus
END_PRESS
```

## 6.2.13 RESOLUTION

### Description

The RESOLUTION method is run through when the current resolution has changed for an opened screen, i.e. typically at a TCU switchover.

### Programming

| | |
|---|---|
| Syntax: | RESOLUTION<br><instructions><br>END_RESOLUTION |
| Description: | Resolution |
| Parameters: | - None - |

### See also

Variables S_RESX and S_RESY (Page 120)

## 6.2.14 RESUME

### Description

The RESUME method is called when the screen was interrupted, e.g. for an area change, and is now shown again. The value changes are processed again, if required, the timers started and the RESUME block executed.

### Programming

| | |
|---|---|
| Syntax: | RESUME<br><instruction><br>END_RESUME |
| Description: | Screen active again |
| Parameters: | - None - |

## 6.2.15 SUSPEND

### Description

The SUSPEND method is called when the screen is interrupted and not unloaded. This is the case, for example, when the screen is exited by a simple area change, but not explicitly unloaded. The screen is retained in the background, but no value change or timers are processed. The screen is paused.

### Programming

| | |
|---|---|
| Syntax: | SUSPEND<br><instruction><br>END_SUSPEND |
| Description: | Interruption of the screen |
| Parameters: | - None - |

### Example

```
SUSPEND
   MyVar1 = MyVar1 + 1
END_SUSPEND
```

## 6.2.16 Example: Version management with OUTPUT methods

### Overview

Additional variables can be added to existing dialogs when expanding the user interface. A version identifier in parentheses is appended to the additional variables in the definition following the variable name: (0 = Original, is not written), 1 = Version 1, 2 = Version 2, etc.

**Example:**

```
DEF var100=(R//1)              ; Original, corresponds to Version 0
DEF var101(1)=(S//"Hello")     ; Extension as of Version 1
```

When writing the OUTPUT method, you can specify which variables are written, with reference to a particular version identifier.

**Example:**

```
OUTPUT(NC1)                    ; Only the variables of the original are offered in
                               the OUTPUT method.
```

```
OUTPUT(NC1,1)                    ; The variables of the original and the extensions
                                 with version identifier 1 are offered in the OUTPUT
                                 method.
```

The OUTPUT method for the original does not need a version identifier, however you can specify it with 0. OUTPUT(NC1) is equivalent to OUTPUT(NC1,0). Version identifier n in the OUTPUT method includes all variables of the originals 0, 1, 2, ... up to and including n.

## Programming with version identifier

```
//M(XXX)                         Version 0 (default)
DEF var100=(R//1)
DEF var101=(S//"Hello")
DEF TMP
VS8=("GC")
PRESS(VS8)
   GC("NC1")
END_PRESS

OUTPUT(NC1)
var100",,"var101
END_OUTPUT

; ************ Version 1, extended definition ***************
//M(XXX)
DEF var100=(R//1)
DEF var101=(S//"Hello")
DEF var102(1)=(V//"HUGO")
DEF TMP
VS8=("GC")
PRESS(VS8)
   GC("NC1")
END_PRESS
...

OUTPUT(NC1)                      Original and the new version in addition
var100","var101
END_OUTPUT
...

OUTPUT(NC1,1)                    Version 1
var100","var101"," var102
END_OUTPUT
```

# 6.3 Functions

## Overview

A variety of functions are available in dialogs and dialog-dependent softkey menus. These can be activated by specific events (exit input field, actuate softkey) and configured in methods.

## Subprograms

Repeatedly used configuring instructions or others, which define the process for a particular operation can be configured in subprograms. Subprograms can be loaded into the main program or other subprograms at any time and executed as often as necessary, i.e. the instructions they contain do not need to be configured repeatedly. The definition blocks of the dialogs/softkey menu constitute a main program.

## External functions

Additional, user-specific functions can be integrated by means of external functions. The external functions are stored in a DLL file and identified by an entry in the definition lines of the configuration file.

## PI services

The PI_START function starts PI Services (Program Invocation Services) from the PLC in the NC area.

## See also

Function (FCT) (Page 157)

PI services (Page 171)

## 6.3.1 Reading and writing drive parameters: RDOP, WDOP, MRDOP

## Description

You can read and write drive parameters using RDOP, WDOP and MRDOP functions.

**Note**

Do not read drive parameters faster than in a 1 second cycle, slower is better.

Reason: The communication with the drives can otherwise be extremely disturbed or even cause failures.

**Note**

If errors occur when reading and writing drive parameters, the screen property ERR is set appropriately.

## Programming

| Syntax: | **RDOP** *("Identifier of the drive object", "Parameter number")* | |
|---|---|---|
| Description: | Reading a drive parameter (Drive Object Parameter) | |
| Parameter: | Identifier of the drive object | The identifier of the drive object can be found in the "DO name" column in "Drive system diagnostics" in the diagnostics operating area ( > ETC > HSK 8: drive system) (see diagram below). |
| | Parameter number | |

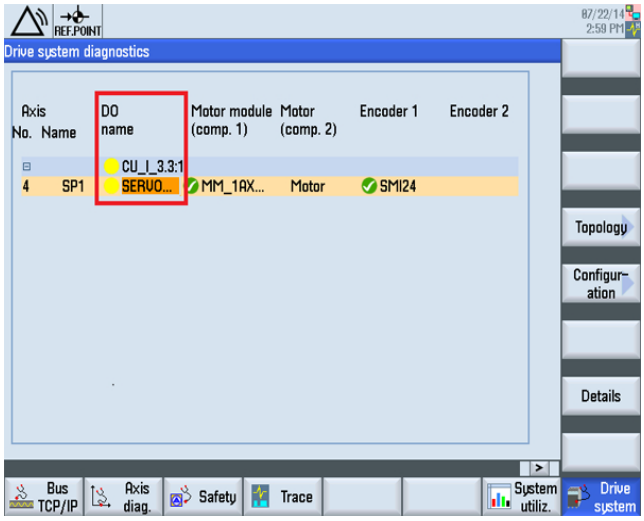



Figure 6-1 Identifier of the drive object

| Syntax: | **MRDOP** *("Identifier of the drive object", "Parameter number1"*"Parameter number"[*…], Register index)* |
|---|---|
| Description: | Multi-read of drive parameters.<br>The MRDOP command transfers several drive parameters or drive objects in a single register access. This access method is significantly faster than reading via individual access attempts. |

| | | |
|---|---|---|
| Parameter: | Identifier of the drive object | The identifier of the drive object can be taken, for example, from the root screen of the "Setup" operating area. |
| | Parameter number1 ... n | In the variable names, "*" is the separator. The values are transferred to register REG[Register index] in the order that the variable names appear in the command. |
| | Register index | The value of the first variable is located in REG[Register index].<br>The value of the second variable is located in REG[Register index + 1] |

| | | |
|---|---|---|
| Syntax: | **WDOP***("Identifier of the drive object", "Parameter number", Value)* | |
| Description: | Writing a drive parameter (Drive Object Parameter) | |
| Parameter: | Identifier of the drive object | The identifier of the drive object can be taken, for example, from the root screen of the "Setup" operating area. |
| | Parameter number | Parameter number |
| | Value | Value to be written |

### Examples

Read motor temperature r0035 of drive object "SERVO_3.3:2":

```
MyVar=RDOP("SERVO_3.3:2","35") ;
```

Read motor temperature r0035 and actual torque value r0080 of drive object "SERVO_3.3:2" and store the respective results as of register index 10:

```
MRDOP("SERVO_3.3:2","35*80",10)
```

## 6.3.2 Subprogram call (CALL)

### Description

The CALL function calls a loaded subprogram from any point in a method. Subprogram nesting is supported, i.e. you can call a subprogram from another subprogram.

### Programming

| Syntax: | **CALL***("Identifier")* | |
|---|---|---|
| Description: | Subprogram call | |
| Parameter: | Identifier | Name of subprogram |

### Example

```
//M(SCREEN FORM1)
DEF VAR1 = ...
DEF VAR2 = ...
CHANGE(VAR1)
  ...
  CALL("MY_UP1")              ; Call subprogram and execute
  ...
END_CHANGE

CHANGE(VAR2)
  ...
  CALL("MY_UP1")              ; Call subprogram and execute
  ...
END_CHANGE

SUB(MY_UP1)
                              ;Do something
END_SUB
//END
```

## 6.3.3 Define block (//B)

### Description

In the program file, subprograms are identified by the block identifier //B and terminated with // END. Several subprograms can be defined under each block identifier.

**Note**

The variables used in the subprogram must be defined in the dialog in which the subprogram is called.

### Programming

A block is structured in the following way:

| | | |
|---|---|---|
| Syntax: | **//B**(*Block name*)<br>**SUB***(Identifier)*<br>END_SUB<br>[SUB(*Identifier*)<br>...<br>END_SUB]<br>...<br>//END | |
| Description: | Defines a subprogram | |
| Parameters: | Block name | Name of block identifier |
| | Identifier | Name of subprogram |

### Example

```
//B(PROG1)                        ; Block start
SUB(UP1)                          ; Start of subprogram
  ...
  REG[0] = 5                      ; Assign value 5 to register 0
  ...
END_SUB                           ; End of subprogram
SUB(UP2)                          ; Start of subprogram
  IF VAR1.val=="Otto"
    VAR1.val="Hans"
    RETURN
  ENDIF
  VAR1.val="Otto"
END_SUB                           ; End of subprogram
//END                             ; Block end
```

## 6.3.4 Check Variable (CVAR)

### Description

You can use the CVAR (CheckVariable) function to run a scan to ascertain whether all or only certain variables or Help variables in a screen form are error-free.

It may be useful to check if variables contain a valid value before an NC code with the GC function.

A variable is error-free if the state of the variable Identifier.vld = 1.

### Programming

| Syntax: | **CVAR**(*VarN*) | |
|---|---|---|
| Description: | Checks variables for valid content | |
| Parameters: | VarN | List of variables to be checked.<br>Up to 29 variables, each separated by a comma, can be checked. A character length of 500 must not be exceeded.<br>The result of the scan can be: |
| | 1 = | TRUE (all variables have valid content) |
| | 0 = | FALSE (at least one variable has invalid content) |

### Example

```
IF CVAR == TRUE                          ; Check all variables
  VS8.SE = 1                             ; If all variables are error-free, soft-
                                         key VS8 is visible
ELSE
  VS8.SE = 2                             ; If a variable has an invalid value,
                                         softkey VS8 is disabled
ENDIF

IF CVAR(”VAR1”, ”VAR2”) == TRUE
                                         ; Check variables VAR1 and VAR2
  DLGL ("VAR1 and VAR2 are OK")
                                         ; If the values of VAR1 and VAR2 are er-
                                         ror-free, "VAR1 and VAR2 are OK" appears
                                         in the dialog line
ELSE
  DLGL ("VAR1 and VAR2 are not OK")
                                         ; If the values of VAR1 and VAR2 are in-
                                         valid, "VAR1 and VAR2 are not OK" appears
                                         in the dialog line
ENDIF
```

## 6.3.5 CLEAR_BACKGROUND

### Description

The CLEAR_BACKGROUND function deletes the LINE, RECT, ELLIPSE, V_SEPARATOR and H_SEPARATOR graphic elements.

### See also

Line, dividing line, rectangle, circle and ellipse (Page 193)

## 6.3.6 Copy Program file function (CP)

### Description

The CP (Copy Program) function copies files within the HMI file system or within the NC file system.

### Programming

| Syntax: | **CP***("Source file", "Target file")* | |
|---|---|---|
| Description: | Copies a file | |
| Parameter: | Source file | Complete path to the source file |
| | Target file | Complete path data of the target file |

The return value (VAR1 defined as help variable) queries whether the function was successful:

```
CP("//NC/MPF.DIR/HOHO.MPF","//NC/MPF.DIR/ASLAN.MPF",VAR1)
```

### Example

Application with return value:

```
CP("//NC/MPF.DIR/HOHO.MPF","//NC/MPF.DIR/ASLAN.MPF",VAR1)
CP("CF_CARD:/wks.dir/MESS_BILD.WPD/MESS_BILD.MPF","//NC/WKS.DIR/AAA.WPD/HOHO2.MPF",VAR1)
CP("//NC/MPF.DIR/HOHO.MPF","//NC/MPF.DIR/ASLAN.MPF")                    ; WPD must exist
```

Application without return value:

```
CP("//NC/MPF.DIR/HOHO.MPF","//NC/MPF.DIR/ASLAN.MPF")
CP("CF_CARD:/mpf.dir/MYPROG.MPF","//NC/MPF.DIR/HOHO.MPF")
CP("//NC/MPF.DIR/HOHO.MPF","CF_CARD:/XYZ/MYPROG.MPF")                   ; XYZ must exist
```

### See also

Support of FILE_ERR: FILE_ERR variable (Page 112)

## 6.3.7 Delete Program file function (DP)

### Description

The DP (Delete Program) function deletes a file from the passive HMI or active NC file system.

### Programming

| | | |
|---|---|---|
| Syntax: | **DP***("File")* | |
| Description: | Delete file | |
| Parameter: | File | Complete path name of file to be deleted |

### Example

The following data management syntax is used for this function:

- With return value

  ```
  DP("//NC/MPF.DIR/MYPROG.MPF",VAR1)
  DP("//NC/WKS.DIR/TEST.WPD/MYPROG.MPF",VAR1)
  DP("//NC/CMA.DIR/MYPROG.SPF",VAR1)
  ```

  VAR1 = 0 File was deleted.

  VAR1 = 1 File was not deleted.
- Without return value:

  ```
  DP("//NC/MPF.DIR/MYPROG.MPF")
  DP("//NC/WKS.DIR/TEST.WPD/MYPROG.MPF")
  DP("//NC/CMA.DIR/MYPROG.SPF")
  ```

## 6.3.8 Exist Program file function (EP)

### Description

The EP (Exist Program) function checks whether a particular NC program is stored on the specified path in the NC or HMI file system.

### Programming

| | | |
|---|---|---|
| Syntax: | **EP***("File")* | |
| Description: | Checks the existence of the NC program | |
| Parameter: | File | Complete path to the file in the NC or HMI file system |
| Return value: | Name of a variable to which the result of the scan should be assigned | |

The EP function can handle the new syntax and the old logic (with adapted syntax).

The file is directly addressed using a qualifying name:

```
//NC/MPF.DIR/XYZ.MPF
```

or
`CF_CARD: /MPF.DIR/XYZ.MPF` (points to /user/sinumerik/data/prog)
or
`LOC:` (corresponds to CF_CARD)

**Examples of new syntax:**

```
EP("//NC/WKS.DIR/TEST.WPD/XYZ.MPF",VAR1)
EP("CF_CARD:/mpf.dir/XYZ.MPF",VAR1)
EP("LOC:/mpf.dir/XYZ.MPF",VAR1)
; With return value:
; VAR1 = 0 File exists.
; VAR1 = 1 File does not exist.
```

**Example of old syntax:**

```
EP("/MPF.DIR/CFI.MPF", VAR1)
; With return value:
; VAR1 = M File is located in the HMI file system.
; VAR1 = N File is located in the NC file system.
; VAR1 = B File is located in the HMI and the NC file system.
```

## Example

```
EP("/MPF.DIR/GROB.MPF",VAR1)                                    ; Old - path now with / instead of \

IF VAR1 == "M"
   DLGL("File is located in the HMI file system")
ELSE
   IF VAR1 == "N"
      DLGL("File is located in the NC file directory")
   ELSE
     DLGL("File is located neither in the HMI nor in the NC
file system")
   ENDIF
ENDIF
```

## 6.3.9 Move Program file function (MP)

## Description

The MP (Move Program) function copies files within the HMI file system or within the NC file system.

### Programming

| Syntax: | **MP***("source", "target")* | |
|---|---|---|
| | `MP("CF_CARD:/MPF.DIR/MYPROG.MPF","//NC/MPF.DIR")` | |
| Description: | Move file | |
| Parameter: | Source file | Complete path data |
| | Target file | Complete path data |
| Return value | Result of the query | |

### Examples

With return value:

```
MP("//NC/MPF.DIR/123.MPF","//NC/MPF.DIR/ASLAN.MPF",VAR1)          ;Within NC
MP("//NC/MPF.DIR/123.MPF",VAR0,VAR1)                              ;Target via variable
MP(VAR4,VAR0,VAR1)                                                ;Source and target via variable
MP("CF_CARD:/mpf.dir/myprog.mpf","//NC/MPF.DIR/123.MPF",VAR1)     ;From CF card to NC
MP("//NC/MPF.DIR/HOHO.MPF","CF_CARD:/XYZ/123.mpf",VAR1)           ;From NC to CF card

; With return value:
; VAR1 = 0 Executed
; VAR1 = 1 Not executed
```

## 6.3.10 Select Program file function (SP)

### Description

The SP (Select Program) function selects a file in the active NC file system for execution, i.e. the file must be loaded into the NC beforehand.

### Programming

| Syntax: | **SP***("File")* | |
|---|---|---|
| Designation: | Select program | |
| Parameter: | "File" | Complete path name of NC file |

### Example

The following data management syntax is used for this function:

- With return value

  `SP("//NC/MPF.DIR/MYPROG.MPF", VAR1)`

  VAR1 = 0 File was loaded.

VAR1 = 1 File was not loaded without return value.

- Without return value:

```
SP("//NC/MPF.DIR/MYPROG.MPF")
```

## 6.3.11 File accesses: RDFILE, WRFILE, RDLINEFILE, WRLINEFILE

### Description

The RDFILE and WRFILE functions are available for read and write access to files with INI syntax.

The RDLINEFILE and WRLINEFILE functions are available for read and write access to individual lines of a file.

### Programming

| Syntax: | **RDFILE**(*"File name + path", "Section", "Key"*) | |
|---|---|---|
| Description: | Reading from a file | |
| Parameter: | File name + path | Path and file name |
| | Section | Section in the INI file |
| | Key | Key in the INI file |

| Syntax: | **WRFILE***(Value, "File name + path", "Section", "Key")* | |
|---|---|---|
| Description: | Writing to a file | |
| Parameter: | Value | Value to be written |
| | File name + path | Path and file name |
| | Section | Section in the INI file |
| | Key | Key in the INI file |

| Syntax: | **RDLINEFILE***("File name + path", Line number)* | |
|---|---|---|
| Description: | Reading a line from a file | |
| Parameter: | File name + path | Path and file name |
| | Line number | Line number<br>Numbering begins at 0 for the first line. |

| Syntax: | **WRLINEFILE***(Value, "File name + path")* |
|---|---|
| Description: | Writing of a line at the end of a file |

| Parameter: | Value | Value to be written |
|---|---|---|
| | File name + path | Path and file name |

**Note**

- The files must not be in the file system of the NC (data management).
- If the file does not exist, or the end of the file is reached or other errors occur, the variables FILE_ERR and ERR are set accordingly. You can check whether a file exists first with the Exist Program (EP) file function.
- The processed file is generated in "UTF-8" coding (without BOM (Byte Order Mask). When reading a file, it is expected in "UTF-8" coding.
- You can delete the file explicitly when required with the Delete Program (DP) file function.

## Examples

**Reading from an INI file:**

Requirement/assumption:

Content of file "C:/tmp/myfile.ini":

<…>
```
[MyData]
MyName=Daniel
```
<…>

```
MyVar = RDFILE("C:/tmp/myfile.ini", "MyData", "MyName")
```

Result:

MyVar now contains the value "Daniel".

**Writing to an INI file:**

Requirement/assumption:

VARS=12

```
WRFILE(VARS, "C:/tmp/myfile.ini", "MySession", "NrOfSessions")
```

Result:

Content of file "C:/tmp/myfile.ini":

<…>
```
[MySession]
NrOfSessions=12
```
<…>

**Reading the 4th line of a file:**

Requirement/assumption:

Content of c:/tmp/myfile.mpf:

```
R[0]=0
F3500 G1
MYLOOPIDX1: X0 y-50 Z0
  X150 Y50 Z10
  R[0]=R[0]+1
GOTOB MYLOOPIDX1
M30
```

```
MyVar = RDLINEFILE("C:/tmp/myfile.mpf", 4)
```

Result:

MyVar now contains the value "  R[0]=R[0]+1"

**Writing at the end of a file:**

Requirement/assumption:

```
VARX=123
VARY=456
```

```
WRLINEFILE("F100 X" << VARX << " Y" << VARY, "C:/tmp/mypp.mpf")
```

Result:

Content of c:/tmp/mypp.mpf:

```
<...>
F100 X123 Y456
```

## 6.3.12 Dialog line (DLGL)

### Description

It is possible to configure short texts (messages or input tips) for output in the dialog line of the dialog in response to certain situations.

Possible number of characters in the default font size: approx. 50

### Programming

| Syntax: | **DLGL***("String")* | |
|---|---|---|
| Description: | Outputs text in the dialog line | |
| Parameter: | String | Text, which is displayed in the dialog line |

### Example

```
IF Var1 > Var2
                              ; The text "Value too large!" appears in the dialog
                              line if variable1 > variable2.
  DLGL("Value too large!")
ENDIF
```

## 6.3.13 DEBUG

### Description

The DEBUG function provides an analysis help during the draft phase of Run MyScreen user screens. An expression in brackets transferred with the DEBUG function is evaluated during runtime. The result is attached to the "easyscreen_log.txt" logbook as a separate entry.

Each entry has the current time stamp in square brackets as prefix (see example below).

It is recommended that the DEBUG outputs are removed in time-critical parts when possible. Particularly after completing the draft phase, it is recommended that you comment out the DEBUG outputs. The write access to the constantly expanding "easyscreen_log.txt" logbook may slow down the processing.

### Programming

| Syntax: | **DEBUG***(expression)*) |
|---|---|
| Description: | Makes an entry in the "easyscreen_log.txt" logbook |
| Parameter: | Expression to be evaluated from which an entry is generated in the logbook |

### Example

```
IF Var1 > Var2
  DEBUG("Value of ""Var1"": " << Var1)
                                        ; Entry in the "easyscreen_log_txt:
                                        [10:22:40.445] DEBUG: Value of "Var1":
                                        123.456
ENDIF
```

## 6.3.14 Exit dialog (EXIT)

### Description

The EXIT function is used to exit a dialog and return to the master dialog. If no master dialog is found, you will exit the newly configured user interfaces and return to the standard application.

### Programming (without parameters)

| | |
|---|---|
| Syntax: | **EXIT** |
| Description: | Exits a dialog |
| Parameters: | - None - |

### Example

```
PRESS(HS1)
  EXIT
END_PRESS
```

### Description

If the current dialog has been called with a transfer variable, the value of the variables can be changed and transferred to the output dialog.

The variable values are each assigned to the variables transferred from the output dialog to the subsequent dialog using the "LM" function. Up to 20 variable values, each separated by a comma, can be transferred.

**Note**

The sequence of variables or variable values must be the same as the sequence of transfer values programmed for the LM function to preclude assignment errors. Any unspecified variable values will not be changed when the transfer is made. The modified transfer variables are immediately valid in the output dialog on execution of the LM function.

### Programming with a transfer variable

| | | |
|---|---|---|
| Syntax: | **EXIT***[(VARx)]* | |
| Description: | Exits the dialog and transfers one or more variables | |
| Parameters: | VARx | Label variables |

### Example

```
//M(Screen form1)
...
PRESS(HS1)
  LM("SCREEN FORM2","CFI.COM",1, POSX,
POSY, DIAMETER)
                                        ; Interrupt screen form1 and open screen
                                        form2. In doing this, transfer variables
                                        POSX, POSY and DIAMETER.
  DLGL("Screen form2 ended")            ; After returning from screen form2, the
                                        following text appears in the dialog line
                                        of screen form1: Screen form2 ended.
END_PRESS
...
//END

//M(Screen form2)
...
PRESS(HS1)
EXIT(5, , CALCULATED_DIAMETER)
                                        ; Exit screen form2 and return to screen
                                        form1 in the line after LM. In doing
                                        this, assign the value 5 to the variable
                                        POSX and the value of the CALCULATED_DI-
                                        AMETER variable to the DIAMETER variable.
                                        The variable POSY retains its current val-
                                        ue.
END_PRESS
...
//END
```

## 6.3.15 Dynamic manipulation of the lists of toggle fields or list box fields

### Description

The LISTADDITEM, LISTINSERTITEM, LISTDELETEITEM and LISTCLEAR functions are used for the dynamic manipulation of the lists of toggle fields or list box fields.

These functions only have an effect on variables that have their own list, such as:

- "Simple" list
  `DEF VAR_AC1 = (I/* 0,1,2,3,4,5,6,7,8)` or
- "Extended" list
  `DEF VAR_AC2 = (I/* 0="AC0", 1="AC1", 2="AC2", 3="AC3", 4="AC4", 5="AC5", 6="AC6", 7="AC7", 8="AC8").`

If the variable points to an array, e.g. `DEF VAR_AC3 = (I/* MYARRAY)`, these functions are not available as otherwise the global array would be changed.

At least one value must be defined for the variable in the DEF line. This specifies the "Simple" or "Extended" list type.

However, the list can then be completely deleted and, if required, completely rebuilt. The "Simple" or "Extended" type must be retained however, it cannot be changed dynamically.

## Programming

| Syntax: | **LISTINSERTITEM***(Variable name, Position, ItemValue[, ItemDisp-Value])* | |
|---|---|---|
| Description: | Inserts an element at a specific position; reading from a file | |
| Parameter: | Variable name | |
| | Position | Position at which an element is to be inserted in the list |
| | ItemValue | Value of the list entry |
| | ItemDispValue | Value as it is to be displayed in the list |

| Syntax: | **LISTADDITEM**(Variable name, ItemValue[, ItemDispValue]) | |
|---|---|---|
| Description: | Adds an element to the end of the list | |
| Parameter: | Variable name | |
| | ItemValue | Value of the list entry |
| | ItemDispValue | Value as it is to be displayed in the list |

| Syntax: | **LISTDELETEITEM**(Variable name, Position) | |
|---|---|---|
| Description: | Deletes a specific element | |
| Parameter: | Variable name | |
| | Position | Position of the element to be deleted in the list |

| Syntax: | **LISTCOUNT**(Variable name) | |
|---|---|---|
| Description: | Returns the current number of elements in the list | |
| Parameter: | Variable name | |

| Syntax: | **LISTCLEAR***(Variable name)* | |
|---|---|---|
| Description: | Deletes the complete list | |
| Parameter: | Variable name | |

## Examples

**Note**

The following examples build upon each other. The sequence of the examples is relevant in order to understand the respective results.

Requirement/assumption:

```
DEF VAR_AC = (I/* 0="Off",1="On"/1/,"Switch"/WR2)
```

**Adding an element -1="Undefined" to the end of the list:**

```
LISTADDITEM("VAR_AC", -1, """Undefined""")
```

Result: `0="Off", 1="On", -1="Undefined"`

**Inserting an element 99="Maybe" at position 2:**

```
LISTINSERTITEM("VAR_AC", 2, 99, """Maybe""")
```

Result: `0="Off", 1="On", 99="Maybe", -1="Undefined"`

**Determining the current number of list elements:**

```
REG[10]=LISTCOUNT("VAR_AC")
```

Result: REG[10] = 4

**Deleting the element at position 1:**

```
LISTDELETEITEM("VAR_AC", 1)
```

Result: `0="Off", 99="Maybe", -1="Undefined"`

**Deletion of the complete list:**

```
LISTCLEAR("VAR_AC")
```

Result: List is empty

## 6.3.16 Evaluate (EVAL)

### Description

The EVAL function evaluates a transferred expression and then executes it. With this function, expressions can be programmed during runtime. This can be useful, for example, for indexed access operations to variables.

### Programming

| Syntax: | **EVAL***(exp)* | |
|---|---|---|
| Description: | Evaluates an expression | |
| Parameters: | exp | Logic expression |

### Example

```
VAR1=(S)
VAR2=(S)
VAR3=(S)
VAR4=(S)
CHANGE()
  REG[7] = EVAL("VAR"<<REG[5])  ; The expression in parentheses produces VAR3 if the
                                value of REG[5] is equal to 3. The value of VAR3 is,
                                therefore, assigned to REG[7].
  IF REG[5] == 1
    REG[7] = VAR1
  ELSE
    IF REG[5] == 2
      REG[7] = VAR2
    ELSE
      IF REG[5] == 3
        REG[7] = VAR3
      ELSE
        IF REG[5] == 4
          REG[7] = VAR4
        ENDIF
      ENDIF
    ENDIF
  ENDIF
END_CHANGE
```

## 6.3.17 Exit Loading Softkey (EXITLS)

### Description

You can use the EXITLS function to exit the current user interface and load a defined softkey menu.

### Programming

| Syntax: | **EXITLS***("Softkey menu"[, "Path*"]) | |
|---|---|---|
| Description: | Exits dialog and loads a softkey menu | |
| Parameters: | Softkey menu | Name of the softkey menu to be loaded |
| | Path name | Directory path of the softkey menu to be loaded |

### Example

```
PRESS(HS1)
    EXITLS( "Menu1", "AEDITOR.COM" )
END_PRESS
```

## 6.3.18 Function (FCT)

### Description

The external functions are stored in a DLL file and identified by an entry in the definition lines of the configuration file.

**Note**

The external function must have at least one return parameter.

### Programming

| Syntax: | **FCT***Function name = ("File"/Type of return/Types of permanent parameters/ Types of variable parameters)* |
|---|---|
| | FCT InitConnection = ("c:\tmp\xyz.dll"/I/R,I,S/I,S) |
| Description: | An external function can, for example, be called from the LOAD method or executed in the PRESS method. |

| | | |
|---|---|---|
| Parameters: | Function name | Name of external function |
| | File | Complete path to DLL file |
| | Type of return | Data type of the return value |
| | Type of fixed parameter | Value parameter |
| | Type of variable parameter | Reference parameter |
| | The data types are separated by commas. | |

The external function can, for example, be called from the LOAD method or executed in the PRESS method.

**Example:**

```
press(vs4)
RET = InitConnection(VAR1,13,"Servus",VAR2,VAR17)
end_press
```

## Structure of the external function

The external function must take into account a certain, specific signature:

| | | |
|---|---|---|
| Syntax: | extern "C" dllexport void InitConnection (ExtFctStructPtr FctRet, ExtFctStructPtr FctPar, char cNrFctPar) | |
| Description: | DLL export, only when implemented in Windows<br>Specifiers and transfer parameters are strictly defined. The actual call parameters are transferred using the transferred structures. | |
| Parameters: | cNrFctPar | Number of call parameters = number of structure elements in FctPar |
| | FctPar | Pointer to a field of structure elements, which contain the particular call parameter with data type. |
| | FctRet | Pointer to a structure for the function value return with data type. |

## Definition of the transfer structure

```
union CFI_VARIANT
    (
    char                                b;
    short int                           i;
    double                              r;
    char*                               s;
    )
typedef struct ExtFctStructTag
    (
```

```
    char                                cTyp;
    union CFI_VARIANT                   value;
    )ExtFctStruct;
typedef struct ExtFct* ExtFctStructPtr;
```

If the external function is to be developed independently of the platform (Windows, Linux), then it is not permissible to use the keyword __declspec(dllexport). This keyword is only required under Windows. For instance, the following macro can be used under Qt:

```
#ifdef Q_WS_WIN
       #define MY_EXPORT __declspec(dllexport)
#else
       #define MY_EXPORT
#endif
```

The function is declared as follows:

```
extern "C" MY_EXPORT void InitConnection
(ExtFctStructPtr FctRet, ExtFctStructPtr FctPar,char cNrFctPar)
```

If the screens, configured with "Run MyScreens", are used on the NCU and PCU/PC, then the extension of the binary file must be omitted:

```
FCT InitConnection = ("xyz"/I/R,I,S/I,S)
```

When the absolute path information is omitted, "Run MyScreens" first searches for the binary file in the configured directory.

## 6.3.19 Generate code (GC)

### Description

The GC (Generate Code) function generates NC code from the OUTPUT method.

### Programming

| | |
|---|---|
| Syntax: | **GC**("*Identifier*"[,"*Target file*"][,*Opt*],[*Append*]) |
| Description: | Generating NC code (the GC function can only be used within the NC) |

| | | |
|---|---|---|
| Parameter: | Identifier | Name of the OUTPUT method from which code is generated |
| | Target file | Path name of target file for HMI or NC file system<br>If the target file is not specified (only possible within Programming Support), the code will be written to the location of the cursor within the file that is currently open. |
| | Opt | Option for generating comments. |
| | 0: | (Default setting) Generate code with comment for the purpose of recompilability (see also Recompile (Page 176)). |
| | 1: | Do not create comments in the generated code.<br>**Note:** This code cannot be recompiled (see also Recompile (Page 176)). |
| | Append | This parameter is only relevant if a target file is specified. |
| | 0: | (Default setting) If the file already exists, the old content is deleted. |
| | 1: | If the file already exists, the new code is written at the start of the file. |
| | 2: | If the file already exists, the new code is written at the end of the file. |

## Example

```
//M(TestGC/"Code generation:")
DEF VAR1 = (R//1)
DEF VAR2 = (R//2)
DEF D_NAME
LOAD
  VAR1 = 123
  VAR2 = -6
END_LOAD
OUTPUT(CODE1)
  "Cycle123(" VAR1 "," VAR2 ")"
  "M30"
END_OUTPUT

PRESS(VS1)
  D_NAME = "\MPF.DIR\MESSEN.MPF"
  GC("CODE1",D_NAME)                ; Write NC code from the OUTPUT method to file
                                    \MPF.DIR\MESSEN.MPF:
                                    Cycle123(123, -6)
                                    M30
END_PRESS
```

## Storing the target file

The GC function can only be used within the NC. Use the function CP or MP to move the file. See the following examples:

```
; Windows directory
  GC("Code","//NC/MPF.DIR/NC1.MPF")
  MP("//NC/MPF.DIR/NC1.MPF","d:/tmp/WIN1.MPF")

; LOCAL_DRIVE
  GC("Code","//NC/MPF.DIR/NC1.MPF")
  MP("//NC/MPF.DIR/NC1.MPF","LOCAL_DRIVE:/WIN_LD1.MPF")
```

## Recompile

- **No entry for target file:**
  The GC function can only be used in the Programming Support and writes the NC code to the file currently open in the editor. Recompilation of the NC code is possible. If the GC function is configured without a target file being specified under "Run MyScreens", an error message is output when it is executed.
- **Entry for target file:**
  The code generated from the OUTPUT method is entered in the target file. If the target file does not already exist, it is set up in the NC file system. If the target file is stored in the HMI file system, it is stored on the hard disk. User comment lines (information required to recompile code) are not set up, i.e. the code cannot be recompiled.

## Special features of recompilation

The GC function cannot be called in sub-dialogs because variables originating from master dialogs can be used in sub-dialogs. These variables would not, however, be available in response to a direct call.

When generated code is processed manually with the editor, the number of characters for values created by the code generation program must not be changed. Changing these values would make it impossible to recompile the code.

Remedy:

1. Recompile
2. Make change using the configured dialog (e.g. 99 → 101)
3. GC

## 6.3.20 Password functions

### Overview

The following password functions are available:

- Set password
- Delete password
- Change password

### HMI_LOGIN function: Set new password

| Syntax: | **HMI_LOGIN***(Passwd)* | |
|---|---|---|
| Description: | A password which sets the current access level is sent to the NCK with the HMI_LOGIN() function. | |
| Parameters: | Passwd | Password |

### Example

```
REG[0] = HMI_LOGIN("CUSTOMER")          ; Result = TRUE if the password has been set
                                          successfully, otherwise FALSE
```

### HMI_LOGOFF function: Delete password

| Syntax: | **HMI_LOGOFF** |
|---|---|
| Description: | The actual access level can be reset using the HMI_LOGOFF function. |
| Parameters: | - None - |

### Example

```
REG[0] = HMI_LOGOFF                     ; Result = TRUE if the password has been de-
                                          leted successfully, otherwise FALSE
```

### HMI_SETPASSWD function: Change password

| Syntax: | **HMI_SETPASSWD***(AC, Passwd)* | |
|---|---|---|
| Description: | The password of the current level or a lower level can be changed with the HMI_SETPASSWD function. | |
| Parameters: | AC | Access level |
| | Passwd | Password |

### Example

```
REG[0] = HMI_SETPASSWD(4,"MYPWD")      ; Result = TRUE if the password has been
                                       changed successfully, otherwise FALSE
```

## 6.3.21 Load Array (LA)

### Description

The LA (Load Array) function can be used to load an array from another file.

### Programming

| Syntax: | **LA***(Identifier [, File])* | |
|---|---|---|
| Description: | Loads an array from a file | |
| Parameters: | Identifier | Name of array to be loaded |
| | File | File in which the array is defined |

**Note**

If an array in the current configuration file must be replaced by an array from another configuration file, then both arrays must have the same name.

### Example

```
                                       ; Extract from file  maske.com
DEF VAR2 = (S/*ARR5/"Out"/,"Toggle field")
PRESS(HS5)
  LA("ARR5","arrayext.com")            ; Load array ARR5 from file arrayext.com
  VAR2 = ARR5[0]                       ; "Above"/"Below"/"Right"/"Left" appears
                                       in the VAR2 toggle field
                                       instead of "Out/In"
END_PRESS
//A(ARR5)
("Out"/"In")
//END
                                       ; Extract from file arrayext.com
//A(ARR5)
("Above"/"Below"/"Right"/"Left")
//END
```

**Note**

Please note that a valid value must be assigned to a variable after the LA function has been used to assign another array to the toggle field of the variable.

## 6.3.22 Load Block (LB)

### Description

The LB (Load Block) function can be used to load blocks containing subprograms during runtime. LB should be configured in a LOAD method so that the loaded subprograms can be called at any time.

**Note**

Subprograms can also be defined directly in a dialog so that they do not have to be loaded.

### Programming

| | | |
|---|---|---|
| Syntax: | **LB***("Block name"[,"File"])* | |
| Description: | Loads subprogram during runtime | |
| Parameters: | Block name | Name of block identifier |
| | File | Path name of configuration file<br>Default setting = Current configuration file |

### Example

```
LOAD
  LB("PROG1")              ; Block "PROG1" is searched for in the current configura-
                           tion file and then loaded.
  LB("PROG2","XY.COM")     ; Block "PROG2" is searched for in the configuration file
                           XY.COM and then loaded.
END_LOAD
```

## 6.3.23 Load Mask (LM)

### Description

The LM function loads a new dialog. Depending on the mode of the dialog change (see table below), a message dialog can also be implemented with this function.

### Master dialog / sub-dialog

A dialog, which calls another dialog, but is not ended itself, is referred to as a master dialog. A dialog that is called by a master dialog is referred to as a sub-dialog.

### Programming

| | | |
|---|---|---|
| Syntax: | **LM***("Identifier"[,"File"] [,MSx [, VARx] ] )* | |
| Description: | Loads a dialog | |
| Parameter: | Identifier | Name of the dialog to be loaded |
| | File | Path name (HMI file system or NC file system) of the configuration file; default setting: Current configuration file |
| | MSx | Mode of dialog change |
| | 0: | (Default setting) The current dialog is rejected, the new dialog is loaded and displayed. EXIT will send you back to the standard application. You can use the MSx parameter to determine whether or not the current dialog should be terminated when changing dialogs. If the current dialog is retained, variables can be transferred to the new dialog.<br>The advantage of the MSx parameter is that the dialogs do not always need to be reinitialized when they are changed. Instead, the data and layout of the current dialog are retained and data transfer is made easier. |
| | 1: | The current master dialog is interrupted when the LM function is initiated. The new sub-dialog is loaded and displayed (e.g. for the implementation of a message dialog). EXIT will end the sub-dialog and return to the point at which the master dialog was interrupted.<br>In the master dialog, the UNLOAD block is not processed during the interruption. |
| | VARx | Precondition: MS1<br>List of variables which can be transferred from the master dialog to the sub-dialog. Up to 20 variables, each separated by a comma, can be transferred. |

**Note**

Parameter VARx transfers only the value of the variable in each case, i.e. variables can be read and written in the sub-dialog, but are not visible in it. Variables can be returned from the sub-dialog to the master dialog by means of the EXIT function.

### Example

**SCREEN FORM1 (master dialog): Jump to the sub-dialog SCREEN FORM2 with variable transfer**

```
PRESS(HS1)

  LM("SCREEN FORM2","CFI.COM",1, POSX, POSY, DIAMETER)
                                    ; Interrupt screen form1 and open screen form2:
                                    In doing this, variables POSX, POSY and DIAME-
                                    TER are transferred.
  DLGL("Screen form2 ended")        ; After returning from screen form2, the fol-
                                    lowing text appears in the dialog line of
                                    screen form1: Screen form2 ended.
END_PRESS
```

**SCREEN FORM2 (sub-dialog): Return to master dialog SCREEN FORM1 with transfer of the variable contents**

```
PRESS(VS8)

  EXIT(POSX, POSY, DIAMETER)
                                    ; Hide screen form2 and continue screen form1:
                                    In doing this, the changed variables POSX, POSY
                                    and DIAMETER are transferred.
END_PRESS
```

## 6.3.24 Load Softkey (LS)

### Description

The LS function can be used to display another softkey menu.

### Programming

| Syntax: | **LS***("Identifier"[, "File"][, Merge])* |
|---|---|
| Description: | Displays the softkey menu |

| | | |
|---|---|---|
| Parameters: | Identifier | Name of the softkey menu |
| | File | Path (HMI file system or NC file system) to the configuration file<br>Default setting: Current configuration file |
| | Merge | |
| | 0: | All existing softkeys are deleted; the newly configured softkeys are entered. |
| | 1: | Default setting<br>Only the newly configured softkeys overwrite the available softkeys. The other softkeys (= softkeys of the HMI application) are kept with their functionality and text. |

## Example

```
PRESS(HS4)
  LS("Menu2",,0)      ; Menu2 overwrites the existing softkey menu, the softkeys
                      that are displayed are deleted.
END_PRESS
```

**Note**

As long as the Interpreter has not displayed a dialog, i.e. no LM function has yet been processed, only one LS or one LM command, but no other action, can be configured in the PRESS method of the definition block for the start softkey and the softkey menu.

The LS and LM functions may only be called within a softkey PRESS method and will not react if navigation keys are pressed (PU, PD, SL, SR, SU, SD).

## 6.3.25 Load Grid (LG)

### Description

The table description (grid) can be dynamically provided within methods (e.g. LOAD) using the LG method.

In order to assign a table using the LG method, the variable must have already been defined as a grid variable and cross-referenced to an existing, valid table.

### Programming

| | |
|---|---|
| Syntax: | **LG** *(Grid name, Variable name [,File name])* |
| Description: | Loads a table |

| | | |
|---|---|---|
| Parameters: | Grid name | Name of the table (grid) in inverted commas |
| | Variable name | Name of the variable to which the table is to be assigned, in inverted commas |
| | File name | Name of the file in which the table (grid) is defined, in inverted commas. Only needs to be specified if the table is not defined within the file that also contains the definition of the variable. |

### Example

```
//M(MyGridSample/"My Grid Sample")
DEF MyGridVar=(R/% MyGrid1///WR2////100,,351,100)
LOAD
   LG("MyGrid1","MyGridVar","mygrids.com")
END_LOAD
```

**Content of mygrids.com:**

```
//G(MyGrid1/0/5)
(I///,"MyGrid1"/wr1//"1"/80/1)
(R3///"LongText1","R1-R4"/wr2//"$R[1]"/80/1)
(IBB///"LongText2","M2.2-M2.5"/wr2//"M2.2"/80/,1)
(R3///"LongText3","R9,R11,R13,R15"/wr2//"$R[9]"/110/2)
//END
```

## 6.3.26 Multiple selection SWITCH

### Description

You can check a variable for different values with a SWITCH command.

The expression configured in the SWITCH statement is compared in succession to all the values configured in the following CASE statements.

If not the same, a comparison is made with the following CASE statement. If a DEFAULT statement follows and up to now no CASE statement was the same as the expression configured in the SWITCH statement, the statements following the DEFAULT statement are executed.

If the same, the following statements are executed until a CASE, DEFAULT or END_SWITCH statement occurs.

### Example

```
FOCUS
   SWITCH (FOC)
      CASE "VarF"
         DLGL("Variable ""VarF"" has the input focus.")
      CASE "VarZ"
         DLGL("Variable ""VarZ"" has the input focus.")
      DEFAULT
         DLGL("Any other variable has the input focus.")
   END_SWITCH
END_FOCUS
```

## 6.3.27 Multiple Read NC PLC (MRNP)

### Description

This MRNP command transfers several NC/PLC variables in a single register access. This access method is significantly faster than reading via individual access attempts. The NC/PLC variables must be included within an MRNP command of the same area.

The areas of the NC/PLC variables are organized as follows:

- General NC data ($MN..., $SN.., /nck/...)
- Channel-specific NC data ($MC..., $SC.., /channel/...)
- PLC data (DB..., MB.., /plc/...)
- Axis-specific NC data on the same axis ($MA..., $SA..)

### Programming

| | |
|---|---|
| Syntax: | **MRNP***(Variable name 1*Variable name 2[* ...], Register index)* |
| Description: | Reads several variables |
| Parameter: | In the variable names, "*" is the separator. The values are transferred to register REG[Register index] and those following in the order that the variable names appear in the command.<br>The following therefore applies:<br>The value of the first variable is located in REG[Register index].<br>The value of the second variable is located in REG[Register index + 1], etc. |

**Note**

It should be noted that the number of registers is restricted and the list of variables cannot exceed 500 characters.

## Example

```
MRNP("$R[0]*$R[1]*$R[2]*$R[3]",1)      ;The values of variables $R[0] to $R[3] are
                                        written to REG[1] to REG[4].
```

## NC variable

All machine data, setting data and R parameters are available, but only certain system variables (see also: List of accessible system variables (Page 311)).

All global and channel-specific user variables (GUDs) can be accessed. However, local and program-global user variables cannot be processed.

| Machine data | |
|---|---|
| Global machine data | $MN_... |
| Axis-specific machine data | $MA_... |
| Channel-specific machine data | $MC_... |

| Setting data | |
|---|---|
| Global setting data | $SN_... |
| Axis-specific setting data | $SA_... |
| Channel-specific setting data | $SC_... |

| System variables | |
|---|---|
| R parameter 1 | $R[1] |

## PLC variable

All PLC data are available.

| PLC data | |
|---|---|
| Byte y bit z of data block x | DBx.DBXy.z |
| Byte y of data block x | DBx.DBBy |
| Word y of data block x | DBx.DBWy |
| Double word y v. of data block x | DBx.DBDy |
| Real y of data block x | DBx.DBRy |
| Flag byte x bit y | Mx.y |

| PLC data | |
|---|---|
| Flag byte x | MBx |
| Flag word x | MWx |
| Flag double word x | MDx |
| Input byte x bit y | Ix.y or Ex.y |
| Input byte x | IBx or EBx |
| Input word x | IWx or EWx |
| Input double word x | IDx or EDx |
| Output byte x bit y | Qx.y or Ax.y |
| Output byte x | QBx or ABx |
| Output word x | QWx or AWx |
| Output double word x | QDx or ADx |
| String y with length z from data block x | DBx.DBSy.z |

## 6.3.28 PI services

### Description

The PI_START function starts PI Services (Program Invocation Services) from the PLC in the NC area.

**Note**

A list of the available PI services can be found in the Basic Functions Function Manual.

### Programming

| Syntax: | **PI_START**("*Transfer string*") | |
|---|---|---|
| Description: | Executes the PI service | |
| Parameter: | "Transfer string" | Unlike the OEM documentation, the transfer string should be entered in inverted commas. |

### Example

```
PI_START("/NC,001,_N_LOGOUT")
```

**Note**

Channel-dependent PI Services always refer to the current channel.

PI services of the tool functions (TO area) always refer to the TO area that is assigned to the current channel.

## 6.3.29 Reading (RNP) and writing (WNP) system or user variables

### Description

The RNP (Read NC PLC) command can be used to read NC or PLC variables or machine data.

### Programming

| Syntax: | **RNP** ("*System or user variable", value*) | |
|---|---|---|
| Description: | Reads NC or PLC variable or machine data | |
| Parameters: | System or user variable | Name of NC or PLC variable |
| | Value | Value that is to be written to the system or user variable.<br>If the value is a String type, it must be written in double quotation marks. |

### Example

```
VAR2=RNP("$AA_IN[2]")         ; Read NC variable
```

### Description

The WNP (Write NC PLC) command can be used to write NC or PLC variables or machine data.

NC/PLC variables are accessed anew every time the WNP function is executed, i.e., NC/PLC access is always executed in a CHANGE method. It is advisable to use this option in cases where a system or user variable changes value frequently. If an NC/PLC variable is to be accessed only once, then it must be configured in a LOAD or UNLOAD method.

### Programming

| Syntax: | **WNP***("System or user variable", value)* | |
|---|---|---|
| Description: | Writes NC or PLC variable or machine data | |
| Parameters: | System or user variable | Name of NC or PLC variable |
| | Value | Value that is to be written to the system or user variable.<br>If the value is a String type, it must be written in double quotation marks. |

### Example

```
WNP("DB20.DBB1",1)         ; Write PLC variable
```

## 6.3.30 RESIZE_VAR_IO and RESIZE_VAR_TXT

### Description

The `RESIZE_VAR_IO()` and `RESIZE_VAR_TXT()` commands changes the geometry of the input/output component or text component of a variable.

After setting the new geometry, all fields of the screen are positioned as if the screen had been configured with these positions from the start. In this way, all fields are correctly aligned, depending on the configuration.

### Programming

| Syntax: | **RESIZE_VAR_IO***(Variable name, [X], [Y], [Width], [Height])*<br>**RESIZE_VAR_TXT***(Variable name, [X], [Y], [Width], [Height])* | |
|---|---|---|
| Description: | Changes the geometry of the input/output component or text component of a variable. | |
| Parameter: | Variable name | Name of the variable whose geometry of the input/output component or text component of a variable is to be changed. |
| | X | X-coordinate, top left |
| | Y | Y-coordinate, top left |
| | Width | Width |
| | Height | Height |

**Note**

The previous value is retained for each value of X, Y, Width or Height that is not specified. If -1 is specified for one of these values, "Run MyScreens" sets the specified component of the default position.

### Example

```
RESIZE_VAR_IO("MyVar1", 200, , 100)    ; The I/O component of the "MyVar1" variable
                                       is moved to the X position 200 pixels, the
                                       width is set to 100 pixels. The Y position and
                                       the height of the previous value are kept.
```

## 6.3.31 Register (REG)

### Register description

Registers are needed in order to exchange data between different dialogs. Registers are assigned to each dialog. These are created when the first dialog is loaded and assigned the value 0 or an empty string.

**Note**

Registers must not be used directly in an OUTPUT method for generating NC code.

### Programming

| | | |
|---|---|---|
| Syntax: | **REG***[x]* | |
| Description: | Defines a register | |
| Parameters: | x | Register index with x = 0...19;<br>Type: REAL or STRING = VARIANT<br>Registers with x ≥ 20 have already been assigned by Siemens. |

### Description of register value

The assignment of values to registers is configured in a method.

**Note**

If a new dialog is generated from an existing dialog by means of the LM function, register content is automatically transferred to the new dialog at the same time and is available for further calculations in the second dialog.

### Programming

| | | |
|---|---|---|
| Syntax: | *Identifier*.val = *Register value*<br>or<br>*Identifier* = *Register value* | |
| Description: | | |
| Parameters: | Identifier | Name of the register |
| | Register value | Value of register |

### Example

```
UNLOAD
  REG[0] = VAR1          ; Assign value of variable 1 to register 0
```

```
END_UNLOAD

UNLOAD
  REG[9].VAL = 84        ; Assign value 84 to register 9
END_UNLOAD
                         ; These registers can then be assigned to local variables
                         again in a method in the next dialog.
LOAD
  VAR2 = REG[0]
END_LOAD
```

### Description of register status

The Status property can be used to scan a register for valid content.

One possible use for the register scan function is to ensure that a value is written to a register only if the relevant dialog is a "master dialog".

### Programming

| Syntax: | *Identifier*.vld | |
|---|---|---|
| Description: | The property is read-only property. | |
| Parameters: | Identifier | Name of the register |
| Return value: | | The result of the scan can be: |
| | FALSE = | invalid value |
| | TRUE = | valid value |

### Example

```
IF REG[15].VLD == FALSE    ; Scan validity of register value
  REG[15] = 84
ENDIF
VAR1 = REG[9].VLD          ; Assign the value of the REG[9] status request to Var1.
```

## 6.3.32 RETURN

### Description

The RETURN function can be used to prematurely terminate execution of the current subprogram and to return to the branch point of the last CALL command.

If no RETURN command is configured in the subprogram, the subprogram will run to the end before returning to the branch point.

### Programming

| Syntax: | **RETURN** | |
|---|---|---|
| Description: | Returns to the branch point | |
| Parameters: | - None - | |

### Example

```
//B(PROG1)                ; Block start
SUB(UP2)                  ; Start of subprogram
  IF VAR1.val=="Otto"
    VAR1.val="Hans"
    RETURN                ; If the variable value = Otto, the value "Hans" is as-
                          signed to the variable, and the subprogram ends at this
                          point.
  ENDIF
  VAR1.val="Otto"         ; If the variable value ≠ Otto, the value "Otto" is as-
                          signed to the variable.
END_SUB                   ; End of subprogram
//END                     ; Block end
```

## 6.3.33 Recompile

### Description

In the programming support system, it is possible to **recompile** NC code that has been generated with the GC function and to display the variable values in the input/output field of the associated entry dialog again.

### Programming

Variables from the NC code are transferred to the dialog. At the same time, the variable values from the NC code are compared with the calculated variable values from the configuration file. If the values do not coincide, an error message is written to the log book because values have been changed during NC code generation.

If the NC code contains the same variable several times, it is evaluated at the point where it last occurs during recompilation. A warning is also written to the log book.

Variables not utilized in NC code during code generation are stored as user comment. The term "user comment" refers to all information required to recompile codes. User comment must not be altered.

**Note**

The block consisting of NC code and user comment can be recompiled only if it starts at the beginning of a line.

**Examples:**

The programm contains the following NC code:

```
DEF VAR1=(I//101)
OUTPUT(CODE1)
  "X" VAR1 " Y200"
  "X" VAR1 " Y0"
END_OUTPUT
```

The following code is then stored in the parts program:

```
;NCG#TestGC#\cus.dir\aeditor.com#CODE1#1#3#
X101 Y200
X101 Y0
;#END#
```

The Editor reads the following during recompilation:

```
X101 Y200
X222 Y0          ; The value for X has been changed in the part program (X101 → X222)
```

The following value is displayed for VAR1 in the input dialog: `VAR1 = 222`

## See also

Generate code (GC) (Page 159)

## 6.3.34 Recompile without user comment

### Description

In the Programming Support, it is possible to **recompile without user comments** the NC code that has been generated with the GC function and to display the variable values in the input/ output field of the associated entry dialog again.

## Programming

The GC command can be executed in the following way in order to suppress comment lines that are generated for standard code generation:

```
GC("CODE1",D_NAME,1)
```

Normally, the resulting code cannot be recompiled. The following steps are required in order to be able to recompile the cycle calls generated in this way:

- **Extend the file "easyscreen.ini"**
  The section `[RECOMPILE_INFO_FILES]` is introduced into the "easyscreen.ini" file. In this section, all ini files are listed that contain descriptions for cycles recompiled without user comment:
  ```
  [RECOMPILE_INFO_FILES]
  IniFile01 = cycles1.ini
  IniFile02 = cycles2.ini
  ```
  Several ini files can be specified, whose names can be freely selected.
- **Creating an ini file for a cycle description**
  Save the ini file with the cycle descriptions under the following path:
  *[System user directory]*/cfg
  *[System oem directory]*/cfg
  *[System add_on directory]*/cfg
  A separate section is required for each cycle. The section name corresponds to the name of the cycle:
  ```
  [Cycle123]
  Mname = TestGC
  Dname = testgc.com
  OUTPUT = Code1
  Anzp = 3
  Version = 0
  Code_type = 1
  Icon = cycle123.png
  Desc_Text = This is describing text
  ```

| | |
|---|---|
| Mname | Screen form name |
| Dname | Name of the file in which the screen is defined |
| OUTPUT | Name of the relevant OUTPUT method |
| Anzp | Number of parameters of the screen to be recompiled (all with DEF-created variables, also help variables) |
| Version | (optional) version specification for cycle |

| | |
|---|---|
| Icon | (optional) icon for display in the machining step program, format *.png<br>Screen size for corresponding resolution:<br>640 X 480 mm → 16 x 16 pixels<br>800 X 600 mm → 20 x 20 pixels<br>1024 X 768 mm → 26 x 26 pixels<br>1280 X 1024 mm → 26 x 26 pixels<br>1280 X 768 mm → 26 x 26 pixels<br>Archive: *[system user directory]*ico/ico<resolution><br>**Notes:**<br>• For resolutions of 1280, the folder for 1024 x 768 mm is used (only suitable for machining step programs).<br>• The screen size is not just dependent on the resolution, but also on the font size set in the editor. |
| Desc_Text | (optional) Explanation text for display in the machining step program, max. length of string is 17 characters (only suitable for machining step programs). |
| Param_Text | (optional) Text for the parameter list<br>T=%1 D=%2 |

**Note**

**Notes on Icon, Desc_Text and Param_Text:**

1. The user icon is always displayed in the G code editor. The G code icon is always displayed in the ShopMill/ShopTurn editor. This helps to better distinguish between ShopMill/ ShopTurn steps and NON-ShopMill/ShopTurn steps.
2. The Run MyScreens step is dependent on the editor setting "Display cycles as machining step". This applies to both the G code and the JobShop editor.
   This means that when "Display cycles as machining step" = "YES", the "Desc_Text" of the Run MyScreens configuration is taken; when "NO", the cycle is displayed.
3. Desc_Text and Param_Text
   – can be specified according to language via the $xxxxx notation.
   – The parameters of the cycle call generated can be accessed via %1, %2, etc. This involves the placeholder being replaced by the parameter that is specified after the '%' in the parameter list generated.

## Example

```
//M(TestGC/"Code generation:")
DEF VAR1 = (R//1)
DEF VAR2 = (R//2)
DEF D_NAME
LOAD
  VAR1 = 123
```

```
  VAR2 = -6
END_LOAD
OUTPUT(CODE1)
  "Cycle123(" VAR1 "," VAR2 ")"
  "M30"
END_OUTPUT

PRESS(VS1)
  D_NAME = "\MPF.DIR\MESSEN.MPF"
  GC("CODE1",D_NAME)                          ;Write NC code from the OUTPUT method to
                                              file \MPF.DIR\MESSEN.MPF:
                                              Cycle123(123, -6)
                                              M30

END_PRESS
```

## Supplementary conditions

- The "Recompile without user comment" functionality does not have the full range of functions as "Recompile with user comment".
  Typical cycle calls, such as MYCYCLE(PAR1, PAR2, PAR3, …), are supported during "Recompile without user comment". However, there must not be a user comment in the line of the function call. Optional parameters that are not transferred during the function call and are of the string S type must however be at least specified with empty quotation marks, e.g. "". Otherwise, "Run MyScreens" attempts to fill these parameters using commas so that the "filled cycle call" can then be recompiled.
- Parameters of the string type must not have any commas or semicolons in the string to be transferred.
- During "Recompile without user comment", all variables contained in the OUTPUT method must always be within the brackets so that the "Fill in missing cycle parameters with commas" functionality can take effect.

Example:

Permitted:

```
OUTPUT
    "MYCYCLE(" MYPAR1 "," MYPAR2 "," MYPAR3 ")"
END_OUTPUT
```

Not permitted (variable MYCOMMENT is positioned after the closing bracket.):

```
OUTPUT
    "MYCYCLE(" MYPAR1 "," MYPAR2 "," MYPAR3 ")" MYCOMMENT
END_OUTPUT
```

## 6.3.35 Search Forward, Search Backward (SF, SB)

### Description

The **SF, SB (Search Forward, Search Backward)** function is used to search for a string from the current cursor position in the NC program currently selected in the Editor and to output its value.

### Programming

| | | |
|---|---|---|
| Syntax: | **SF***("String")* | |
| Identifiers: | **S**earch **F**orward: Search forward from the current cursor position | |
| Syntax: | **SB***("String")* | |
| Identifiers: | **S**earch **B**ackward: Search backward from the current cursor position | |
| Parameters: | String | Text to be found |

### Rules governing text search

- A blank must be inserted before and after the search concept unit, consisting of search string and its value, in the currently selected NC program.
- The system does not search for concepts within comment text or other strings.
- The value to be output must be a numerical expression. Expressions in the form of "X1=4+5" are not recognized.
- The system recognizes hexadecimal constants in the form of X1='HFFFF', binary constants in the form of X1='B10010' and exponential components in the form of X1='-.5EX-4'.
- The value of a string can be output if it contains the following between string and value:
  - Nothing
  - Blanks
  - Equality sign

### Example

The following notations are possible:

```
X100 Y200                ; The variable Abc is assigned the value 200
Abc = SB("Y")
X100 Y 200               ; The variable Abc is assigned the value 200
Abc = SB("Y")
X100 Y=200               ; The variable Abc is assigned the value 200
Abc = SB("Y")
```

## 6.3.36 STRING functions

### Overview

The following functions enable strings to be processed:

- Determine length of string
- Find a character in a string
- Extract substring from left
- Extract substring from right
- Extract substring from mid-string
- Replace substring
- Compare strings
- Insert a string in another string
- Remove a string from a string
- Remove spaces (from left or from the right)
- Insert values or strings with formatting identifiers

### LEN function: Length of a string

| Syntax: | **LEN***(string \| varname)* | |
|---|---|---|
| Description: | Determines the number of characters in a string | |
| Parameters: | string | Every valid string expression. NULL is output if string is empty. |
| | varname | Any valid declared variable name |
| | Only one of the two parameters is allowed. | |

### Example

```
DEF VAR01
DEF VAR02

LOAD
  VAR01="HALLO"
  VAR02=LEN(VAR01)          ; Result = 5
END_LOAD
```

### INSTR function: Search for character in string

| Syntax: | **INSTR***(Start, String1, String2 [,Direction])* |
|---|---|
| Description: | Searches for characters |

| | | |
|---|---|---|
| Parameters: | Start | Starting position for searching for string1 in string2. Enter 0 to start searching at the beginning of string2. |
| | String1 | Character that is being searched for. |
| | String2 | Character string in which the search is being made. |
| | Direction (optional) | Direction in which the search is being made.<br>0: From left to right (default setting)<br>1: From right to left |
| | 0 is returned if string1 does not occur in string2. | |

## Example

```
DEF VAR01
DEF VAR02

LOAD
  VAR01="HALLO/WELT"
  VAR02=INST(1,"/",VAR01)        ; Result = 6
END_LOAD
```

## LEFT Function: String from left

| | | |
|---|---|---|
| Syntax: | **LEFT***(string, length)* | |
| Description: | LEFT returns a string containing the specified number of characters starting from the left-hand side of a string. | |
| Parameters: | string | Character string or variable with the character string to be processed |
| | length | Number of characters that are to be read out |

## Example

```
DEF VAR01
DEF VAR02
LOAD
  VAR01="HALLO/WELT"
  VAR02=LEFT(VAR01,5)           ; Result = "HELLO"
END_LOAD
```

## RIGHT function: String from right

| | |
|---|---|
| Syntax: | **RIGHT***(string, length)* |
| Description: | RIGHT returns a string containing the specified number of characters starting from the right-hand side of a string. |

| | | |
|---|---|---|
| Parameters: | string | Character string or variable with the character string to be processed |
| | length | Number of characters that are to be read out |

## Example

```
DEF VAR01
DEF VAR02
LOAD
  VAR01="HALLO/WELT"
  VAR02=LEFT(VAR01,4)         ; Result = "WORLD"
END_LOAD
```

## MIDS function: String from mid-string

| | | |
|---|---|---|
| Syntax: | **MIDS***(string, start [, length])* | |
| Description: | MIDS returns a string containing the specified number of characters starting at the specified position in the string. | |
| Parameters: | string | Character string or variable with the character string to be processed |
| | start | Start from where characters are to be read in the string |
| | length | Number of characters that are to be read out |

## Example

```
DEF VAR01
DEF VAR02
LOAD
  VAR01="HALLO/WELT"
  VAR02=LEFT(VAR01,4,4)           ; Result = "LO/W"
END_LOAD
```

## REPLACE Function: Replacing characters

| | |
|---|---|
| Syntax: | **REPLACE***( string, FindString, ReplaceString [, start [, count ] ] )* |
| Description: | REPLACE replaces a character/string in a string with another character/string. |

| | | |
|---|---|---|
| Parameters: | string | String in which FindString is to be replaced with ReplaceString. |
| | FindString | String to be replaced |
| | ReplaceString | Replacement string (is used instead of the FindString) |
| | start | Starting position for search and replace operations |
| | count | Number of characters that are to be searched from the starting position after FindString |
| Return value: | | |
| | string = Empty string | Copy of string |
| | FindString = Empty string | Copy of string |
| | ReplaceString = Empty string | Copy of string, in which all occurrences of FindString are deleted |
| | start > Len(String) | Empty string |
| | count = 0 | Copy of string |

## STRCMP function: Compare strings

| | | |
|---|---|---|
| Syntax: | **STRCMP***(string1, string2 [, CaseInsensitive])* | |
| Description: | STRCMP compares a character string with another character string. | |
| Parameters: | string1 | Character string that is to be compared with a second character string |
| | string2 | Character string that is to be compared with the first character string |
| | CaseInsensitive | Comparison with/without consideration of upper/lower case:<br>Not specified or FALSE = upper/lower case is considered<br>TRUE = upper/lower case is not considered |

## Example

```
REG[0]=STRCMP("Hugo", "HUGO")              ; Result = 32
                                           ; (<> 0, strings are not identical)

REG[0]=STRCMP("Hugo", "HUGO", TRUE)        ; Result = 0
                                           ; (== 0, stings are identical)
```

## STRINSERT function: Insert string

| | | |
|---|---|---|
| Syntax: | **STRINSERT***(string1, string2 , insert position)* | |
| Description: | STRINSERT inserts a character string at a specific position in a character string. | |
| Parameters: | string1 | Character string in which a character string is to be inserted |
| | string2 | Character string that is to be inserted |
| | insert position | Position at which the character string is to be inserted |

### Example

```
REG[0]=STRINSERT("Hello!", " world", 5)    ; Result = "Hello world!")
```

## STRREMOVE function: Remove string

| Syntax: | **STRREMOVE***(string1, remove positon , count)* | |
|---|---|---|
| Description: | STRREMOVE removes a specific number of characters at a specific position within a character string. | |
| Parameters: | string1 | Character string from which characters are to be removed |
| | remove position | Position at which characters are to be removed from the character string |
| | count | Number of characters to be removed |

### Example

```
REG[0]=STRREMOVE("Hello world!", 5, 6)     ; Result = "Hello!")
```

## TRIMLEFT function: Remove blanks from the left of the string

| Syntax: | **TRIMLEFT***(string1)* | |
|---|---|---|
| Description: | TRIMLEFT removes blanks on the left from a character string. | |
| Parameters: | string1 | Character string from which blanks are to be removed from the left of the character string |

### Example

```
REG[0]=TRIMLEFT("    Hello!")              ; Result = "Hello!")
```

## TRIMRIGHT function: Remove blanks from the right of the string

| Syntax: | **TRIMRIGHT***(string1)* | |
|---|---|---|
| Description: | TRIMLEFT removes blanks on the right from a character string. | |
| Parameters: | string1 | Character string from which blanks are to be removed from the right of the character string |

### Example

```
REG[0]=TRIMRIGHT("Hello!    ")             ; Result = "Hello!")
```

## FORMAT function: Insert values or string with formatting identifier

| | | |
|---|---|---|
| Syntax: | **FORMAT***(Text with formatting identifier [, Value1] ... [, Value28] )* | |
| Description: | Through the use of formatting identifiers, the FORMAT function enables up to 28 values or strings to be inserted at specified positions in a specific text. | |
| Formatting identifiers: | Syntax | %[Flags] [Width] [.decimal places] type |
| | Flags | Optional character for defining output formatting:<br>• Right-justified or left-justified ("-") for left-justified<br>• Add leading zeros ("0")<br>• default: Fill in with blanks if the value to be output has fewer positions than specified with [Width]. |
| | Width | The argument defines the minimum output width for a non-negative number. If the value has fewer positions than specified by the argument, the missing spaces are filled with blanks. |
| | Decimal places | With floating-point numbers, the optional parameter defines the number of decimal places. |
| | Type | The type character defines which data formats are transferred to the print statement.<br>These characters need to be specified.<br>• d: Integer value<br>• f: Floating-point number<br>• s: String<br>• o: Octal<br>• x: Hexadecimal<br>• b: Binary |

## Example

```
DEF VAR1
DEF VAR2
LOAD
VAR1 = 123
VAR2 = FORMAT("Hello %08b %.2f %s!", VAR1 + 1, 987.654321, "world")
                                    ; Result = "Hello 01111100 987.65 world!"
END_LOAD
```

## See also

Use of strings (Page 106)

## 6.3.37 WHILE/UNTIL loops

### Description

A loop can be implemented with the DO LOOP commands. Depending on the configuration, it is run through as long as a condition is satisfied (WHILE) or until a condition occurs (UNTIL).

As loops can impair the system performance, depending on the configuration, they should be used carefully and without time-intensive actions in the loops.

It is recommended that a register (REG[]) be used as run variable, because normal display variables (particularly those with system or user variables connection) also impair the system performance due to the frequent updates or write operations.

The runtime of "Run MyScreens" methods can be determined with the DEBUG function (see Section DEBUG (Page 151)). Problems caused by loops (high CPU load, reduced response capability) may be identified in this way.

**Note**

As each FOR loop can be replaced by a WHILE loop, the syntax to formulate a FOR loop is not supported in EasyScreen.

### Programming

```
DO
   <instructions>
LOOP_WHILE <Condition to continue the loop>

DO
   <instructions>
LOOP_UNTIL <Condition to terminate the loop>

DO_WHILE <Condition to continue the loop>
   <instructions>
LOOP

DO_UNTIL <Condition to terminate the loop>
   <instructions>
LOOP
```

### Example

```
  REG[0] = 5
DO
```

```
   DEBUG("OUTER: " << REG[0])
   REG[0] = REG[0] + 1
     REG[1] = -5

   DO
      DEBUG("INNER: " << REG[1])
      REG[1] = REG[1] + 1
   LOOP_WHILE REG[1] < 0

LOOP_WHILE REG[0] < 10
```

```
  REG[0] = 5
DO
   DEBUG("OUTER: " << REG[0])
   REG[0] = REG[0] + 1
     REG[1] = -5

   DO
      DEBUG("INNER: " << REG[1])
      REG[1] = REG[1] + 1
   LOOP_UNTIL 0 <= REG[1]

LOOP_WHILE 10 > REG[0]
```

```
  REG[0] = 5
DO_WHILE 10 > REG[0]
   DEBUG("OUTER: " << REG[0])
   REG[0] = REG[0] + 1
     REG[1] = -5

   DO_UNTIL 0 <= REG[1]
     DEBUG("INNER: " << REG[1])
     REG[1] = REG[1] + 1
   LOOP

LOOP
```

```
  REG[0] = 5
DO_WHILE 10 > REG[0]
```

```
   DEBUG("OUTER: " << REG[0])
   REG[0] = REG[0] + 1
     REG[1] = -5

   DO
     DEBUG("INNER: " << REG[1])
     REG[1] = REG[1] + 1
   LOOP_UNTIL 0 <= REG[1]

LOOP
```

## 6.3.38 Cyclic execution of scripts: START_TIMER, STOP_TIMER

### Description

SUB methods can be called cyclically with the aid of timers. The START_TIMER() and STOP_TIMER() functions are available for this purpose.

**Note**

Only one timer can be configured for each SUB method.

### Programming

| Syntax: | **START_TIMER***("SUB name", Interval)* | |
|---|---|---|
| Description: | Starts cyclic processing of a SUB method | |
| Parameters: | SUB name: | Name of the SUB method to be called cyclically |
| | Interval: | Interval in milliseconds |

| Syntax: | **STOP_TIMER***("SUB name", Interval)* | |
|---|---|---|
| Description: | Stops cyclic processing of a SUB method | |
| Parameters: | SUB name: | Name of the SUB method whose timer is to be stopped. |

### Example

```
//M(TimerSample/"My timer")
DEF MyVariable=(I//0/,"Number of cyclic calls:"/WR1)
VS1=("Start%ntimer")
```

```
//M(TimerSample/"My timer")
VS2=("Stop%ntimer")

SUB(MyTimerSub)
   MyVariable = MyVariable + 1
END_SUB

PRESS(VS1)
   ;Calls SUB "MyTimerSub" every 1000 milliseconds
   START_TIMER("MyTimerSub", 1000)
END_PRESS

PRESS(VS2)
   STOP_TIMER("MyTimerSub")
END_PRESS
```

If START_TIMER is called again for a timer already assigned to a SUB method, then the new interval is taken over if it is different. Otherwise, the second call is ignored.

The smallest interval for the system is100 milliseconds.

If STOP_TIMER is called for a SUB method for which no timer is currently running, the call is ignored.

# Graphic and logic elements 7

## 7.1 Line, dividing line, rectangle, circle and ellipse

### Description

Lines, dividing lines, rectangles and ellipses/circles are configured in the LOAD method:

- Transparent rectangles are created by setting the fill color to the system background color.
- With the ELLIPSE element, a circle is created by setting the height and the width to the same value.
- Horizontal and vertical dividing lines always have the exact window width or window height. No scrollbars are produced by dividing lines.
  If you previously used the RECT element to display dividing lines, e.g. RECT(305,0,1,370,0,0,1), this is detected by the Programming Support and automatically converted to V_SEPARATOR(305,1,"#87a5cd", 1). This applies to both vertical and horizontal dividing lines.

### LINE element

Programming:

| Syntax: | LINE (x1,y1,x2,y2,f,s) | |
|---|---|---|
| Description: | Defines a line | |
| Parameter: | x1 | Start point x-coordinate |
| | y1 | Start point y-coordinate |
| | x2 | End point x-coordinate |
| | y2 | End point y-coordinate |
| | f | Color of the line |
| | s | Line style:<br>1 = solid<br>2 = dashed<br>3 = dotted<br>4 = dashed and dotted |

### RECT element

Programming:

| Syntax: | RECT (x,y,w,h,f1,f2,s) |
|---|---|
| Description: | Defines a rectangle |

| | | |
|---|---|---|
| Parameter: | x | x-coordinate, top left |
| | y | y-coordinate, top left |
| | w | Width |
| | h | Height |
| | f1 | Color of the border |
| | f2 | Fill color |
| | s | Border style:<br>1 = solid<br>2 = dashed<br>3 = dotted<br>4 = dashed and dotted |

## ELLIPSE element

Programming:

| | | |
|---|---|---|
| Syntax: | ELLIPSE(x,y,w,h,f1,f2,s) | |
| Description: | Defines an ellipse or circle | |
| Parameter: | x | x-coordinate, top left |
| | y | y-coordinate, top left |
| | w | Width |
| | h | Height |
| | f1 | Color of the border |
| | f2 | Fill color |
| | s | Border style:<br>1 = solid<br>2 = dashed<br>3 = dotted<br>4 = dashed and dotted |

A circle is created when the values for height and width are the same.

## V_SEPARATOR element

Programming:

| | | |
|---|---|---|
| Syntax: | V_SEPARATOR (x,w,color,pen) | |
| Description: | Defines a vertical dividing line | |
| Parameter: | x | x position |
| | w | Line weight |
| | color | Color |
| | pen | Line style:<br>1 = solid<br>2 = dashed<br>3 = dotted<br>4 = dashed and dotted |

## H_SEPARATOR element

Programming:

| Syntax: | H_SEPARATOR (y,h,color,pen) | |
|---|---|---|
| Description: | Defines a horizontal dividing line | |
| Parameter: | y | y position |
| | h | Line weight |
| | color | Color |
| | pen | Line style:<br>1 = solid<br>2 = dashed<br>3 = dotted<br>4 = dashed and dotted |

## See also

LOAD (Page 130)

## See also

CLEAR_BACKGROUND (Page 143)

# 7.2 Defining an array

## Definition

An array can be used to organize data of the same data type stored in the memory in such a way that it is possible to access the data via an index.

## Description

Arrays can be one- or two-dimensional. A one-dimensional array is treated like a two-dimensional array with just one line or column.

Arrays have start identifier //A and end identifier //END. The number of lines and columns is optional. An array is structured in the following way:

## Programming

| | | |
|---|---|---|
| Syntax: | ***//A****(Identifier)*<br>(a/b...)<br>(c/d...)<br>...<br>//END | |
| Description: | Defines array | |
| Parameters: | Identifier | Name of array |
| | a, b, c, d | Values of array<br>Values of the STRING type must be enclosed in double quotation marks. |

## Example

```
//A(Thread)                    ; Size/lead/core diameter
(0.3 / 0.075 / 0.202)
(0.4 / 0.1   / 0.270)
(0.5 / 0.125 / 0.338)
(0.6 / 0.15  / 0.406)
(0.8 / 0.2   / 0.540)
(1.0 / 0.25  / 0.676)
(1.2 / 0.25  / 0.676)
(1.4 / 0.3   / 1.010)
(1.7 / 0.35  / 1.246)
//END
```

## 7.2.1 Accessing the value of an array element

### Description

The value of an array access operation can be transferred with property Value (identifier.val).

The line index (line number of the array) and the column index (column number of the array) each begin at 0. If a line index or column index is outside the array, the value 0 or a blank string is output and the ERR variable is set to TRUE. Variable ERR is also set to TRUE if a search concept cannot be found.

### Programming

| | | | |
|---|---|---|---|
| Syntax: | Identifier [Z,[M[,C]]].val or<br>Identifier [Z,[M[,C]]] | | |
| Description: | Access to one-dimensional array with only one column | | |
| Syntax: | Identifier [S,[M[,C]]].val] or<br>Identifier [S,[M[,C]]] or | | |
| Description: | Access to one-dimensional array with only one line | | |
| Syntax: | Identifier [Z,S,[M[,C]]].val or<br>Identifier [Z,S,[M[,C]]] | | |
| Description: | Access to two-dimensional array | | |
| Parameters: | Identifier: | Name of array | |
| | Z: | Line value (line index or search concept) | |
| | S: | Column value (column index or search concept) | |
| | M: | Access mode | |
| | | 0 | Direct |
| | | 1 | Searches the line, column directly |
| | | 2 | Searches the column, line directly |
| | | 3 | Searches |
| | | 4 | Searches line index |
| | | 5 | Searches column index |
| | C: | Compare mode | |
| | | 0 | Search concept must be located in the range of values of the line or column. |
| | | 1 | Search concept must be located exactly. |

| | | |
|---|---|---|
| Example: | `VAR1 = MET_G[REG[3],1,0].VAL` | `;Assign Var1 a value from array MET_G` |

### Access mode

- **"Direct" access mode**
  With "Direct" access mode (M = 0), the array is accessed with the line index in Z and the column index in S. Compare mode C is not evaluated.
- **"Search" access mode**
  In the case of access mode M = 1, 2 or 3, the search always commences in line 0 or column 0.

| Mode M | Line value Z | Column value S | Output value |
|---|---|---|---|
| 0 | Line index | Column index | Value from line Z and column S |
| 1 | Search concept:<br>Search in column 0 | Column index of column from which value is read | Value from line found and column S |
| 2 | Line index of line from which return value is read | Search concept:<br>Search in line 0 | Value from line Z and column found |
| 3 | Search concept:<br>Search in column 0 | Search concept:<br>Search in line 0 | Value from line and column found |
| 4 | Search concept:<br>Search in column S | Column index of search column | Line index |
| 5 | Line index of search line. | Search concept:<br>Search in line Z | Column index |

### Compare mode

When compare mode C = 0 is used, the content of the search line or search column must be sorted in ascending order. If the search concept is smaller than the first element or greater than the last, the value 0 or a blank string is output and the error variable ERR is set to TRUE.

When compare mode C = 1 is used, the search concept must be present in the search line or search column. If the search concept cannot be found, the value 0 or a blank string is output and the error variable ERR is set to TRUE.

## 7.2.2 Example Access to an array element

### Prerequisite

Two arrays are defined below. These are the basis for the following examples:

```
//A(Thread)
                (0.3 / 0.075 / 0.202)
                (0.4 / 0.1   / 0.270)
                (0.5 / 0.125 / 0.338)
                (0.6 / 0.15  / 0.406)
                (0.8 / 0.2   / 0.540)
```

```
                (1.0 / 0.25  / 0.676)
                (1.2 / 0.25  / 0.676)
                (1.4 / 0.3   / 1.010)
                (1.7 / 0.35  / 1.246)
//END


//A(Array2)
                ("DES" /     "PTCH" /    "CDM" )
                (0.3 /       0.075 /     0.202 )
                (0.4 /       0.1 /       0.270 )
                (0.5 /       0.125 /     0.338 )
                (0.6 /       0.15 /      0.406 )
                (0.8 /       0.2 /       0.540 )
                (1.0 /       0.25 /      0.676 )
                (1.2 /       0.25 /      0.676 )
                (1.4 /       0.3 /       1.010 )
                (1.7 /       0.35 /      1.246 )
//END
```

## Examples

- **Access mode example 1:**
  The search concept is in Z. This key is always sought in column 0. The value from column S is output with the line index of the concept found.
  ```
  VAR1 = Thread[0.5,1,1]   ;VAR1 has the value 0.125
  ```
  Explanation:
  Search for value 0.5 in column 0 of "Thread" array and output the value found in column 1 of the same line.

- **Access mode example 2:**
  The search concept is in S. This concept is always searched for in line 0. The value from line Z is output with the column index of the concept found:
  ```
  VAR1 = ARRAY2[3,"PTCH",2]   ;VAR1 has the value 0.125
  ```
  Explanation:
  Search for column containing "PTCH" in line 0 of array "Array2". Output the value from the column found and the line with index 3.

- **Access mode example 3:**
  A search concept is in each of Z and S. The line index is searched for in column 0 with the concept in Z and the column index in line 0 with the concept in S. The value from the array is output with the line index and column index found:
  ```
  VAR1 = ARRAY2[0.6,"PTCH",3]   ;VAR1 has the value 0.15
  ```
  Explanation:
  Search for the line with the content 0.6 in column 0 of array "Array2", search for the column with the content "STG" in line 0 of Array2. Transfer the value from the line and column found to VAR1.

- **Access mode example 4:**
  The search concept is in Z. S contains the column index of the column in which concept is being searched for. The line index of the concept found is output:
  ```
  VAR1 = Thread[0.125,1,4]   ;VAR1 has the value 2
  ```
  Explanation:
  Search for value 0.125 in column 1 of array "Thread" and transfer the line index of the value found to VAR1.
- **Access mode example 5:**
  Z contains the line index of line in which concept is being searched for. The search concept is in S. The column index of the concept found is output:
  ```
  VAR1 = Thread[4,0.2,5,1]  ;VAR1 has the value 1
  ```
  Explanation:
  Search in line 4 of the "Thread" array for the value 0.2 and transfer the column index of the value found to VAR1. Comparison mode 1 was selected because the values of line 4 are not sorted in ascending order.

## 7.2.3 Scanning the status of an array element

### Description

The Status property can be used to run a scan to find out whether an array access operation is supplying a valid value.

### Programming

| | |
|---|---|
| Syntax: | *Identifier [Z, S, [M[,C]]]*.vld |
| Description: | Status is a read-only property. |
| Parameters: | Identifier Name of array |
| Return Value: | FALSE =invalid value<br>TRUE = valid value |

### Example

```
DEF MPIT = (R///"MPIT",,"MPIT",""/wr3)
DEF PIT  = (R///"PIT",,"PIT",""/wr3)
PRESS(VS1)
  MPIT = 0.6
  IF MET_G[MPIT,0,4,1].VLD == TRUE
    PIT    = MET_G[MPIT,1,0].VAL
    REG[4] = PIT
    REG[1] = "OK"
  ELSE
    REG[1] = "ERROR"
```

```
  ENDIF
END_PRESS
```

# 7.3 Table description (grid)

## Definition

In contrast to the array, the values of a table (grid) are continually updated. It is a tabular display of values that can come from the NC or PLC. The table is organized in columns and always has variables of the same type in one column.

## Assignment

The reference to a table description is performed in the variables definition:

- The variables definition determines the values to be displayed, and the definition of table elements determines the appearance and arrangement on the screen window. The properties of the input/output fields from the definition line of the variables are taken over in the table.
- The visible area of the table is determined by the width and height of the input/output field. Any lines or columns than cannot be seen can be displayed by scrolling horizontally and vertically.

## Table identifiers

Identifiers of a table containing NCK or PLC values of the same type, which can be addressed via a channel block. The table identifier is differentiated from limits or toggle fields by the addition of a % sign in front of it. The file containing the table description can be specified by adding a comma after the identifier and then inserting the name of the file.

Identifier of a table for values of the same type, e.g. from the NCK or PLC. The table identifier is specified at the position at which the limit values or the toggle field are configured. The table identifier starts with a leading % character. This can be followed by a file name with a comma as separator. It specifies the file in which the table description is defined. Per default, the table is sought in the configuration file in which the screen is configured.

A table definition can also be loaded dynamically, e.g. in the LOAD method via the Load Grid (LG) function.

## System or user variable

This parameter remains empty for tables because the column definition lines contain detailed information about the variables to be displayed. The table description can be provided dynamically.

## Example

Definition of a MyGridVar variable which displays a "MyGrid1" grid in its input/output field (distance from left: 100, distance from top: Standard, width: 350, height: 100).

```
DEF MyGridVar=(V/% MyGrid1///////100,,350,100)
```

### See also

Parameters of variables (Page 92)

Load Grid (LG) (Page 167)

## 7.3.1 Defining a table (grid)

### Description

The table block comprises:

- Header
- 1 to n column descriptions

### Programming

| | | | |
|---|---|---|---|
| Syntax: | **//G***(Table identifier/Table type/Number of lines/ [Fixed line attribute],[Fixed column attribute])* | | |
| Description: | Defines a table (grid) | | |
| Parameters: | Table identifier | The table identifier is used without a leading % sign. It can only be used once in a dialog. | |
| | Table type | 0 (default) | Table for PLC or user data (NCK- and channel-specific data) |
| | | 1 | and others, reserved |
| | Number of lines | Number of lines including header | |
| | | The fixed line or fixed column is not scrolled. The number of columns is the number of columns configured. | |
| | Display of column header line | -1: | The column header line is not displayed |
| | | 0: | The column header line is displayed (default) |
| | Number of fixed columns | 0: | No fixed column |
| | | 1 ... n: | Number of columns that are to be permanently visible, i.e. not scrolled horizontally |

### Examples

| | |
|---|---|
| `//G(grid1/0/5/-1,2)` | Column header line is hidden and two fixed columns |
| `//G(grid1/0/5/,1)` | Column header line is displayed and one fixed column |
| `//G(grid1/0/5)` | Column header line is displayed and no fixed columns |

## 7.3.2 Defining columns

### Description

For tables (grids), it is advisable to use variables with an index. For PLC or NC variables, the index number with one or more indices is of significance.

The values displayed in a table can be modified directly within the restrictions of the rights granted by the attributes and within any defined limits.

### Programming

| | | |
|---|---|---|
| Syntax: | *(Type/Limits/Empty/Long text,column header/Attributes/Help display/ System or user variable/Column width/Offset1, Offset2, Offset3)*<br>See also Extended configuration syntax (Page 43). | |
| Description: | Defines columns | |
| Parameters: | Similar to variables | |
| | Type | Data type |
| | Limits | Limit value MIN, limit value MAX |
| | Long text, column header | |
| | Attributes | |
| | Help display | |
| | System or user variable | As variable, PLC or NC variables should be entered in double quotation marks. |
| | Column width | Entry in pixels. |
| | Offset | The increment sizes to increment each index in order to fill the column are specified in the assigned offset parameter:<br>• Offset1: Step width for the 1st index<br>• Offset2: Step width for the 2nd index<br>• Offset3: Step width for the 3rd index |

### Column header from text file

The column header can be entered as text or text number ($8xxxx) and is not scrolled.

### Modifying column properties

The column properties which can be modified dynamically (written) are:

- Limits (min,max),
- Column header (st),
- Attributes (wr, ac and li),
- Help screens (hlp)

Column properties are modified via the variable identifier in the definition line and the column index (starting at 1).

Example: `VAR1[1].st="Column 1"`

The wr, ac and li attributes can be specified for column definitions.

### See also

Load Grid (LG) (Page 167)

## 7.3.3 Focus control in the table (cell)

### Description

The Row and Col properties can be used to set and calculate the focus within a table:

- Identifier.**Row**
- Identifier.**Col**

### Programming

Each cell in a table has the Val and Vld properties.

In order to read and write cell properties, a line and column index must be specified in addition to the variable identifiers from the definition list.

| | |
|---|---|
| Syntax: | Identifier[Line index, column index].val or<br>Identifier[Line index, column index] |
| Description: | Val properties |
| Syntax: | Identifier[Line index, column index].vld |
| Description: | Vld properties |

### Example

```
Var1[2,3].val=1.203
```

If the line and column indices are not specified, the indices of the focused cell apply. This corresponds to:

```
Var1.Row =2
Var1.Col=3
Var1.val=1.203
```

# 7.4 Custom widgets

## 7.4.1 Defining custom widgets

### Description

User-specific display elements are configured in the dialog using a custom widget.



**Software option**

In order to use custom widgets in dialog boxes, you require the following additional software option:

"SINUMERIK Integrate Run MyHMI /3GL" (6FC5800-0AP60-0YB0)

### Programming

| | | |
|---|---|---|
| Definition: | **DEF** *(name)* | |
| Syntax: | **(W///"","***(library name)***.***(class name)***"/////***a***,***b***,***c***,***d***);** | |
| Description: | W | Defining custom widgets |
| Parameters: | Name | Custom widget name, freely selectable |
| | Library name | Can be freely selected, name of the dll (Windows) or (Linux) library file |
| | Class name | Freely selectable, name of the class function from the previously named library |
| | a, b, c, d | Position and size of the configuration |

### Example

A custom widget is defined in the dialog configuration in the following way:

```
DEF Cus = (W///"","slestestcustomwidget.SlEsTestCustomWidget"/////
20,20,250,100);
```

## 7.4.2 Structure of the custom widget library

### Description

Essentially, the custom widget library contains a defined class. The name of this class must be specified in the dialog configuration in addition to the library names. Starting from library names, "Run MyScreens" accesses a dll file with the same name, e.g.:

```
slestestcustomwidget.dll
```

### Programming

The class definition of the dll file should look like this:

```
#define SLESTESTCUSTOMWIDGET_EXPORT Q_DECL_EXPORT


class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget  : public QWidget
{
   Q_OBJECT
....
public slots:
   bool serialize(const QString& szFilePath, bool bIsStoring);
....
}
```

## 7.4.3 Structure of the custom widget interface

### Description

The library is supplemented by an interface in order to display the custom widget in the dialog. This contains macro definitions with which "Run MyScreens" initiates the custom widget. The interface is available in the form of a cpp file. The file name can be freely selected, e.g.: `sleswidgetfactory.cpp`

### Programming

The interface is defined as follows:

```
#include "slestestcustomwidget.h"     ; The header file for the relevant custom widg-
                                      ets is inserted at the beginning of the file
....
//Makros                              ; Macro definitions are not changed
....
WIDGET_CLASS_EXPORT(SlEsTestCustom-   ; The relevant custom widget is declared at
Widget)                               the end of the file
```

### Example

Content of the file sleswidgetfactory.cpp for a custom widget with the class name SlEsTestCustomWidget":

```
#include <Qt/qglobal.h>
#include "slestestcustomwidget.h"
```

```
////////////////////////////////////////////////////////////////////////
// MAKROS FOR PLUGIN DLL-EXPORT - DO NOT CHANGE
////////////////////////////////////////////////////////////////////////

#ifndef Q_EXTERN_C
#ifdef __cplusplus
#define Q_EXTERN_C    extern "C"
#else
#define Q_EXTERN_C    extern
#endif
#endif

#define SL_ES_FCT_NAME(PLUGIN) sl_es_create_ ##PLUGIN
#define SL_ES_CUSTOM_WIDGET_PLUGIN_INSTANTIATE( IMPLEMENTATION , PARAM) \
    { \
    IMPLEMENTATION *i = new PARAM; \
    return i; \
    }

#ifdef Q_WS_WIN
#   ifdef Q_CC_BOR
#       define EXPORT_SL_ES_CUSTOM_WIDGET_PLUGIN(PLUGIN,PARAM) \
    Q_EXTERN_C __declspec(dllexport) void* \
    __stdcall SL_ES_FCT_NAME(PLUGIN) (QWidget* pParent) \
    SL_ES_CUSTOM_WIDGET_PLUGIN_INSTANTIATE( PLUGIN,PARAM )
#   else
#       define EXPORT_SL_ES_CUSTOM_WIDGET_PLUGIN(PLUGIN,PARAM) \
    Q_EXTERN_C __declspec(dllexport) void* SL_ES_FCT_NAME(PLUGIN) \
    (QWidget* pParent) \
    SL_ES_CUSTOM_WIDGET_PLUGIN_INSTANTIATE( PLUGIN,PARAM )
#   endif
#else
#   define EXPORT_SL_ES_CUSTOM_WIDGET_PLUGIN(PLUGIN,PARAM) \
    Q_EXTERN_C void* SL_ES_FCT_NAME(PLUGIN) (QWidget* pParent) \
    SL_ES_CUSTOM_WIDGET_PLUGIN_INSTANTIATE( PLUGIN,PARAM )
#endif

#define WIDGET_CLASS_EXPORT(CLASSNAME)  \
    EXPORT_SL_ES_CUSTOM_WIDGET_PLUGIN(CLASSNAME,CLASSNAME(pParent))

////////////////////////////////////////////////////////////////////////
// FOR OEM USER - please declare here your widget classes for export
////////////////////////////////////////////////////////////////////////
```

```
WIDGET_CLASS_EXPORT(SlEsTestCustomWidget)
```

### 7.4.4 Interaction between custom widget and dialog box - Automatic data exchange

Custom widgets interact with dialog boxes and can display values or manipulate them.

#### Conditions

Automatic data exchange takes place under the following conditions:

| Condition | Direction |
|---|---|
| When starting or recompiling a dialog | Dialog → custom widget |
| When executing the GC command for generating cycle calls | Custom widget → Dialog |

#### Programming

The following definitions are necessary for the interaction:

**Expansion of the dialog configuration**

| Definition: | **DEF** *(variable)* | |
|---|---|---|
| Syntax: | **(***(type)***//5/"","***(variable)***","/wr2/)** | |
| Variable type: | Type | Standard input field (no grid or toggle) with any data type (no W) |
| Parameters: | Variable | Any designation of a variable for data exchange |
| Input mode: | wr2 | Reading and writing |

**Example**

```
DEF CUSVAR1 = (R//5/"","CUSVAR1",""/wr2/)
```

**Expansion of the class definition**

In the class definition of the custom widgets, a QProperty must be created whose name is identical to the selected variable of the dialog configuration, e.g.:

```
Q_PROPERTY(double CUSVAR1 READ cusVar1 WRITE setCusVar1);
```

**Example**

The class definition of the dll file should look like this:

```
#define SLESTESTCUSTOMWIDGET_EXPORT Q_DECL_EXPORT


class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget  : public QWidget
```

```
{
 Q_OBJECT
 Q_PROPERTY(double CUSVAR1  READ cusVar1    WRITE setCusVar1);
....
....
}
```

## 7.4.5 Interaction between custom widget and dialog box - Manual data exchange

Besides automatic data exchange, manual data exchange is also possible. Data exchange is dynamic, i.e. takes place while the dialog box is running. The following actions are possible:

- Properties of the custom widget can be read and written.
- Methods of the custom widget can be called from the Run MyScreens configuration.
- A response to a particular signal of the custom widget can be implemented in order to call subprograms (SUB) in the Run MyScreens configuration.

### 7.4.5.1 Reading and writing properties

#### Description

The functions ReadCWProperties and WriteCWProperties are provided in the Run MyScreens configuration for reading and writing properties of the custom widgets.

#### Programming

| Syntax: | **ReadCWProperty***("Variablename", "Propertyname")* | |
|---|---|---|
| Description: | Reading a property of a custom widget | |
| Parameters: | Variablename | Name of a dialog box variable to which a custom widget is assigned |
| | Propertyname | Name of the custom widget property to be read |
| Return value: | Current value of the custom widget property | |

| Syntax: | **WriteCWProperty***("Variablename", "Propertyname", "Value")* |
|---|---|
| Description: | Writing the property of a custom widget |

| | | |
|---|---|---|
| Parameters: | Variable name | Name of a dialog box variable to which a custom widget is assigned |
| | Propertyname | Name of the custom widget property to be written |
| | Value | Value to be written to the property of the custom widget |

## Examples

### Example 1:

Read property "MyStringVar" of the custom widget that is linked to dialog box variable "MyCWVar1" and assign the value to register 7.

### Custom widget class declaration:

```
class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget  : public QWidget
{
   Q_OBJECT
   Q_PROPERTY(QString MyStringVar
              READ myStringVar
              WRITE setMyStringVar);
   ...
```

### Dialog box configuration:

```
DEF MyCWVar1 = (W///,"slestestcustomwidget.SlEsTestCustomWidget")
PRESS(VS1)
   REG[7]=ReadCWProperty("MyCWVar1", "MyStringVar")
END_PRESS
```

### Example 2:

Write the result of the calculation "3 + sin(123.456)" into property "MyRealVar" of the custom widget that is linked to dialog box variable "MyCWVar1."

### Custom widget class declaration:

```
class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget  : public QWidget
{
   Q_OBJECT
   Q_PROPERTY(double MyRealVar
              READ myRealVar
              WRITE setMyRealVar);
   ...
```

### Dialog box configuration:

```
DEF MyCWVar1 = (W///,"slestestcustomwidget.SlEsTestCustomWidget")
PRESS(VS1)
   WriteCWProperty("MyCWVar1", "MyRealVar", 3 + sin(123.456))
```

```
DEF MyCWVar1 = (W///,"slestestcustomwidget.SlEsTestCustomWidget")
END_PRESS
```

### 7.4.5.2 Executing a method of the custom widget

#### Description

The function CallCWMethod is available in the Run MyScreens configuration for executing methods of the custom widget.

The custom widget method to be called must have no more than 10 transfer parameters.

The following transfer parameter data formats are supported:

- bool
- uint
- int
- double
- QString
- QByteArray

#### Programming

| | | |
|---|---|---|
| Syntax: | **CallCWMethod***("Variablename", "Methodname[, Argument 0][, Argument 1 … [,Argument 9]")* | |
| Description: | Calling a custom widget method | |
| Parameters: | Variable name | Name of a dialog box variable to which a custom widget is assigned |
| | Method name | Name of the custom widget name to be called |
| | Argument 0 - 9 | Transfer parameter for the custom widget method<br>Supported data formats: see above<br>**Note:** The transfer parameters are always transferred with "ByVal," i.e. only the value is transferred and not, for example, the reference to a variable. |
| Return value: | Return value of the custom widget method<br>The following transfer parameter data formats are supported:<br>• void<br>• bool<br>• uint<br>• int<br>• double<br>• QString<br>• QByteArray<br>**Note**: Even if the data format of the return value of the custom widget method is "void," it must be formally assigned to a variable, for example. | |

## Example

### Custom widget class declaration:

```
class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget  : public QWidget
{
   Q_OBJECT
   public slots:
       void myFunc1(int nValue, const QString& szString, double dValue);
...
```

### Dialog box configuration:

```
DEF MyCWVar1 = (W///,"slestestcustomwidget.SlEsTestCustomWidget")
DEF MyStringVar1 = (S)
DEF MyRealVar = (R)

PRESS(VS3)
   REG[9] = CallCWMethod("MyCWVar1", "myFunc1", 1+7, MyStringVar1, sin(MyRealVar) -
8)
END_PRESS
```

## Note

The custom widget must implement the "serialize" method. Here, you have the option of writing the internal data of a custom widget
to a specified file, or restoring it again. This is especially necessary, if, with the "Run MyScreens" screen open, you change to another operating area and then return again. Otherwise, internal data are lost when redisplaying.

| | | |
|---|---|---|
| Syntax: | public slots:<br>bool **serialize**(const QString& szFilePath, bool bIsStoring); | |
| Description: | Reading and writing internal data and states to and from a file | |
| Parameter: | szFilePath | Name of the file with complete path data, in which the internal data and states of the custom widget are written to – or from which they are to be read.<br>If necessary, the file must create the custom widget itself. |
| | bIsStoring | TRUE = write<br>FALSE = read |

### Example

```
bool SlEsTestCustomWidget::serialize(const QString& szFilePath, bool bIsStoring)
{
   QFileInfo fi(szFilePath);
   bool bReturn = false;
   QDir dir;
   if (dir.mkpath(fi.canonicalPath()))
   {
```

```
bool SlEsTestCustomWidget::serialize(const QString& szFilePath, bool bIsStoring)
      QFile fileData(szFilePath);
      QIODevice::OpenMode mode;

      if (bIsStoring)
      {
         mode = QIODevice::WriteOnly;
      }
      else
      {
         mode = QIODevice::ReadOnly;
      }

      if (fileData.open(mode))
      {
         QDataStream streamData;
         streamData.setDevice(&fileData);

         if (bIsStoring)
         {
            streamData << m_nDataCount << m_dValueX;
         }
         else
         {
            streamData >> m_nDataCount >> m_dValueX;
         }

         streamData.setDevice(0);
            fileData.flush();
            fileData.close();
         bReturn = true;
      }
   }
   return bReturn;
}
```

### 7.4.5.3 Response to a custom widget signal

## Description

In Run MyScreens it is possible to respond to a particular signal (invokeSub()) of the custom widget and call up a subprogram (SUB).

10 global variables, the so-called SIGARG, are available for passing values (custom widget signal -> SUB), which are comparable to registers (REG) in configuration. This is where the values transferred with the custom widget signal are stored.

The following transfer parameter data formats are supported:

- bool
- uint
- int
- double
- QString
- QByteArray

## Programming

### Calling the subroutine:

| | | |
|---|---|---|
| Syntax: | `void invokeSub(const QString& rszSignalName, const QVariantList& rvntList);` | |
| Description: | Custom widget signal with which a Run MyScreens subprogram is called. | |
| Parameters: | rszSignalName | Name of the Run MyScreens subprogram to be called |
| | rvntList | QVariantList array for transferring parameters that are stored in the global parameter SIGARG and which are available in the configuration.<br>Maximum size: 10 elements<br>Data formats supported: see above<br>**Note:** The transfer parameters are always passed "By-Val," i.e. only the value and not, for example, the reference to a variable is passed. |

### Subroutine to be called:

| | | |
|---|---|---|
| Syntax: | `SUB(on_<Variablename>_<Signalname>)`<br>`…`<br>`END_SUB` | |
| Description: | Response to a custom widget signal | |
| Parameters: | Variable name | Name of a dialog box variable to which a custom widget is assigned. |
| | Signal name | Name of the custom widget signal |
| | SIGARG 0 - 9 | Transfer parameters for the custom widget method.<br>Supported data formats: See above<br>**Note**: The transfer parameters are always passed "By-Val," i.e. only the value and not, for example, the reference to a variable is passed. |

## Example

### Customer widget class declaration:

```
class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget  : public QWidget
{
```

```
class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget  : public QWidget
   Q_OBJECT
signals:
   void invokeSub(const QString& szSubName, const QVariantList& vntList);
…
}
```

### Custom widget class:

```
QVariantList vntList;
vntList << 123.456;
emit invokeSub("MySub", vntList);
```

### Dialog box configuration:

```
DEF MyCWVar1 = (W///,"slestestcustomwidget.SlEsTestCustomWidget")
SUB(on_MyCWVar1_MySub)
   DEBUG("SUB(on_MyCWVar1_MySub) was called with parameter: """ << SIGARG[0] <<
""""")
END_SUB
```

### Result "easyscreen_log.txt":

```
[10:22:40.445] DEBUG: SUB(on_MyCWVar1_MySub) was called with parameter: "123.456"
```

# 7.5 SIEsGraphCustomWidget

## 7.5.1 SIEsGraphCustomWidget

### General

Using SIEsGraphCustomWidget, you can display geometrical objects (point, line, square, square with rounded corners, ellipse, arc, text) and curves from interpolation points (e.g. measured values, characteristic).

The objects are organized in one or several contours. These can then be displayed, either individually or combined; they can also be entered, selected and deleted.

Using functions, the objects are added to the currently selected contour, and these are then drawn in this particular sequence. If you wish to draw the object in a specific color, then you must set the appropriate pen color before adding. All objects subsequently added will be drawn in this pen color. In addition to the pen color, you can also influence the pen width and pen style. For closed objects such as square, square with rounded edges and ellipse, before adding the objects you can set a fill color.

Every contour can be brought into various modes:

- Standard display of all object types.
- Points can be connected to create a polyline, and can therefore be displayed as curve/graph.
- The area between points, lines or arcs up to the X axis can be filled in with the currently selected fill color. For instance, you can use this option to visualize residual material when turning.
  If, in this mode, points are displayed in addition as polyline, then the complete area below the curve up to the X axis is filled. The points, lines and arcs can be located in all four quadrants.

You can move along one or several contours using a special cursor mode. To do this, the cursor is set to the first or last point object of a contour or, starting from the actual cursor position is moved either forward or backward within the contour of a point object.

When required, a second Y axis (right) can be displayed with its own scaling. This is coupled to the first Y axis (left) through an offset and a factor.

Based on a specified X coordinate, using a search function, a point previously added to the contour can be found, and if required, the cursor can be set to this point.

You can also configure contours as ring buffer with an adjustable size.

As a result of the serialization, the actual state of the SIEsGraphCustomWidget can be saved in a file in binary form - and also restored.

The SIEsGraphCustomWidget can be operated using the "Pan" gesture (where the view is shifted) and the "Pinch"/"Spread" gestures (zooming in and out of the view).

Using the mouse, you can shift the view (left mouse button + Move), and you can zoom in and out (mouse wheel).

For performance reasons, the display is not automatically refreshed. Depending on the particular application, you can initiate the refresh by calling the appropriate function.

## Example

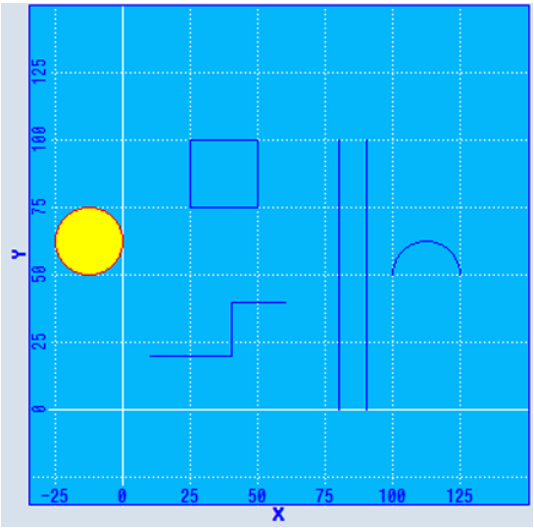



Figure 7-1 SIEsGraphCustomWidget example

```
//M(MyGraphSampleMask/"SlEsGraphCustomWidget Sample")
DEF MyGraphVar = (W///,"slesgraphcustomwidget.SlEsGraphCustomWidget"/////10,10,340,340/0,0,0,0)
VS1=("Add objects",,se1)
LOAD
   WRITECWPROPERTY("MyGraphVar", "AxisNameX", "X")
   WRITECWPROPERTY("MyGraphVar", "AxisNameY", "Y")
   WRITECWPROPERTY("MyGraphVar", "ScaleTextOrientationYAxis", 2)
   WRITECWPROPERTY("MyGraphVar", "KeepAspectRatio", TRUE)

   REG[0]= CALLCWMETHOD("MyGraphVar", "addContour", "MyContour", TRUE)
   REG[0]= CALLCWMETHOD("MyGraphVar", "showContour", "MyContour")
   REG[0]= CALLCWMETHOD("MyGraphVar", "setView", -35, -35, 150, 150)
END_LOAD

PRESS(VS1)
   REG[0]= CALLCWMETHOD("MyGraphVar", "addLine", 10, 20, 40, 20)
   REG[0]= CALLCWMETHOD("MyGraphVar", "addLine", 40, 20, 40, 40)
   REG[0]= CALLCWMETHOD("MyGraphVar", "addLine", 40, 40, 60, 40)
   REG[0]= CALLCWMETHOD("MyGraphVar", "addLine", 80, 0, 80, 100)
   REG[0]= CALLCWMETHOD("MyGraphVar", "addLine", 90, 0, 90, 100)
   REG[0]= CALLCWMETHOD("MyGraphVar", "addRect", 25, 100, 50, 75)
   REG[0]= CALLCWMETHOD("MyGraphVar", "setPenColor", "#ff0000");red
   REG[0]= CALLCWMETHOD("MyGraphVar", "setFillColor", "#ffff00");yellow
   REG[0]= CALLCWMETHOD("MyGraphVar", "addCircle", -12.5, 62.5, 12.5)
   REG[0]= CALLCWMETHOD("MyGraphVar", "setFillColor");off/default
   REG[0]= CALLCWMETHOD("MyGraphVar", "setPenColor");off/default
   REG[0]= CALLCWMETHOD("MyGraphVar", "addArc", 100, 37.5, 125, 62.5, 0, 180)
```

```
//M(MyGraphSampleMask/"SlEsGraphCustomWidget Sample")
   REG[0]= CALLCWMETHOD("MyGraphVar", "update")
END_PRESS
```

## 7.5.2 Notes regarding performance

Depending on the hardware used and the basic utilization of the system, when using SIEsGraphCustomWidgets you must observe the following guide values. In addition to the hardware being used and the basic utilization, the values vary depending on the SIEsGraphCustomWidgets configuration.

Observe these guide values so that the response and stability of the overall system are not diminished.

- Number of SIEsGraphCustomWidgets configured simultaneously (e.g. a maximum of 1)
- Number of configured contours (e.g. a maximum of 6)
- Number of configured graphic objects per contour (e.g. maximum of 1000)
- Frequency of the requested refresh operations (e.g. max. 500 ms)

**Note**

The display is not real-time capable.

## 7.5.3 Reading and writing properties

### Description

The properties listed in the following section are read with ReadCWProperty() and written with WriteCWProperty().

### Examples

Reading the "CursorX" property of the SIEsGraphCustomWidget linked using display variable "MyGraphVar". The result is written to register 0.

```
REG[0] = ReadCWProperty("MyGraphVar", "CursorX")
```

Writes value "MyFirstContour" to the "SelectedContour" property of the SIEsGraphCustomWidget linked using display variable "MyGraphVar".

```
WriteCWProperty("MyGraphVar ", "SelectedContour", "MyFirstContour")
```

## 7.5.4 Properties

### Overview

| Property | Description |
|---|---|
| AxisNameX | Axis identifiers, X axis |
| AxisNameY | Axis identifiers, Y axis |
| AxisNameY2 | Axis identifiers, second Y axis (right) |
| AxisY2Visible | Display/hide second Y axis (right) |
| AxisY2Offset | Offset second Y axis (right) |
| AxisY2Factor | Factor second Y axis (right) |
| ScaleTextEmbedded | Position of the scaling texts |
| ScaleTextOrientationYAxis | Alignment of the scaling texts of the Y axis |
| KeepAspectRatio | Keep aspect ratios |
| SelectedContour | Name of the currently selected contour |
| BackColor | Background color |
| ChartBackColor | Background color of the drawing area |
| ForeColor | Foreground color for text and default pen color |
| ForeColorY2 | Foreground color for the texts of the second Y axis (right) |
| GridColor | Color of the grid lines |
| GridColorY2 | Color of the horizontal grid lines of the second Y axis (right) |
| CursorColor | Color of the cursor crosshairs |
| ShowCursor | Display/hide cursor |
| CursorX | Cursor X position |
| CursorY | Cursor Y position |
| CursorY2 | Cursor Y position referred to the second Y axis (right) |
| CursorStyle | Cursor display type |
| ViewMoveZoomMode | Response when zooming and shifting using gestures |

### AxisNameX – axis identifiers, X axis

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***AxisNameX***")*<br>WriteCWProperty*(GraphVarName, "***AxisNameX***", Value)* | |
| Description: | Reads or sets the identifier of the X axis. If an identifier is not specified, then the space available for the drawing area is fully utilized. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (QString) |
| | Value | Value to be set (QString) |

## AxisNameX – axis identifiers, Y axis

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(*GraphVarName, "**AxisNameY**")<br>WriteCWProperty*(GraphVarName, "***AxisNameY***", Value)* | |
| Description: | Reads or sets the identifier of the Y axis. If an identifier is not specified, then the space available for the drawing area is fully utilized. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (QString) |
| | Value | Value to be set (QString) |

## AxisY2Visible – display/hide second Y axis (right)

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***AxisY2Visible***")*<br>WriteCWProperty*(GraphVarName, "***AxisY2Visible***", Value)* | |
| Description: | Displays/hides second Y axis (right) | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (bool) |
| | Value | Value to be set (bool): TRUE or FALSE |

## AxisY2Offset – offset second Y axis (right)

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***AxisY2Offset***")*<br>WriteCWProperty*(GraphVarName, "***AxisY2Offset***", Value)* | |
| Description: | Offset of the second Y axis (right) referred to the first Y axis (left).<br>See example for AxisY2Factor property. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (double) |
| | Value | Value to be set (double). |

## AxisY2Factor – factor second Y axis (right)

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***AxisY2Factor***")* WriteCWProperty*(GraphVarName, "***AxisY2Factor***", Value)* | |
| Description: | Factor of the second Y axis (right) referred to the first Y axis (left). | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (double) |
| | Value | Value to be set (double). |

Together with the "AxisY2Offset" property, it is possible to display a second Y axis (right) with its own scaling. The scale is coupled coupled to the first Y axis (left) through an offset and a factor.

Formula for converting Y2 to Y:

Y = Y2 / factor - offset

Formula for converting Y to Y2:

Y2 = (Y + offset) * factor

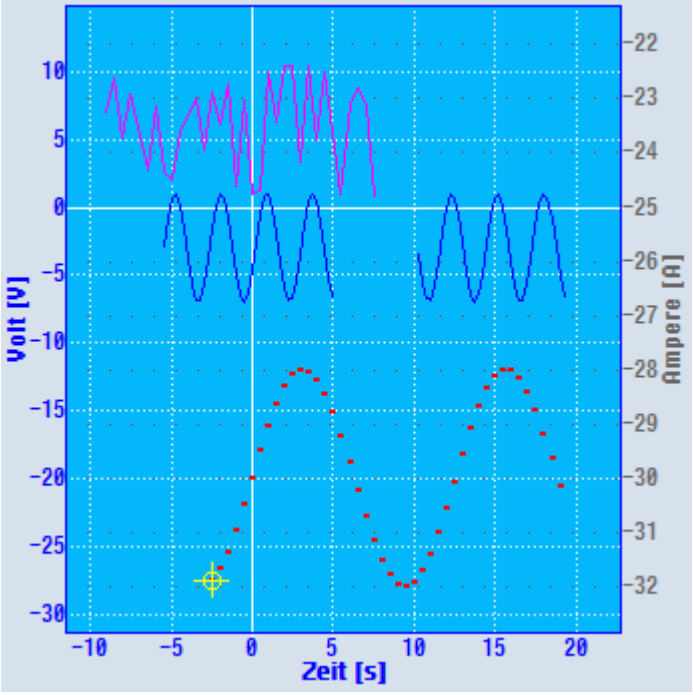



Figure 7-2 Example with second Y axis with own scaling

## Example

Y: 0 to 200 should correspond to Y2: -25 to 25.

- Offset: -100
- Factor: 0.25

**Calculation example:**

Y2 = -30

Y = -30 / 0.25 – (-100) = -20

**Calculation example:**

Y = 0

Y2 = (0 + (-100)) * 0.25 = -25

The X axis is the same for both coordinate systems.

You can set the colors using setForeColorY2() and setGridColorY2() (see colors).

The axes are labeled with setAxisNameY2() (see axis identifiers).

## ScaleTextEmbedded - position of the scaling texts

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***ScaleTextEmbedded***")*<br>WriteCWProperty*(GraphVarName, "***ScaleTextEmbedded***", Value)* | |
| Description: | Using this property you define whether the scaling is positioned inside or outside the drawing area. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (bool) |
| | Value | Value to be set (bool): TRUE or FALSE |

## Example

Scaling text outside the drawing area:

ScaleTextEmbedded = FALSE

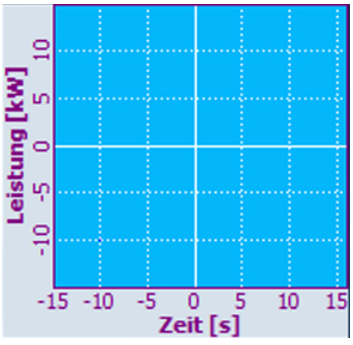



Figure 7-3 Scaling text outside the drawing area

Scaling text within the drawing area:

ScaleTextEmbedded = TRUE

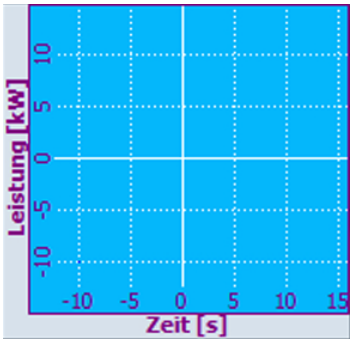



Figure 7-4 Scaling text within the drawing area:

## ScaleTextOrientationYAxis - alignment of the scaling texts of the Y axis

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, '***ScaleTextOrientationYAxis***')*<br>WriteCWProperty*(GraphVarName, '***ScaleTextOrientationYAxis***', Value)* | |
| Description: | Using this function, the scaling texts of the Y axis are aligned vertically. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (int) |
| | Value | Value to be set (int): 1 (= horizontal) or 2 (= vertical) |

## Example

Scaling text within the drawing area:

- ScaleTextEmbedded = TRUE

Horizontal text alignment:

- ScaleTextOrientationYAxis = 1

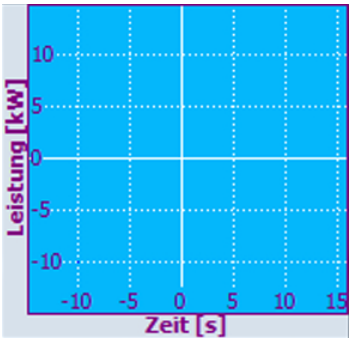



Figure 7-5 Scaling text within the drawing area, horizontal text alignment

Vertical text alignment:

- ScaleTextOrientationYAxis = 2

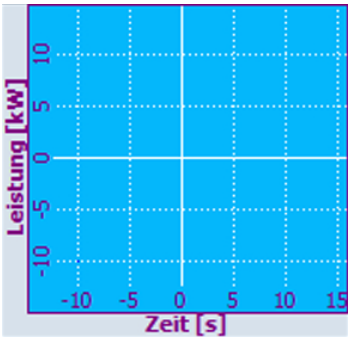



Figure 7-6 Scaling text within the drawing area, vertical text alignment

Scaling text outside the drawing area:

- ScaleTextEmbedded = FALSE

Horizontal text alignment:

- ScaleTextOrientationYAxis = 1

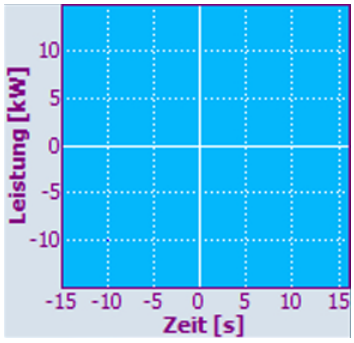



Figure 7-7 Scaling text outside the drawing area, horizontal text alignment

Vertical text alignment:

- ScaleTextOrientationYAxis = 2

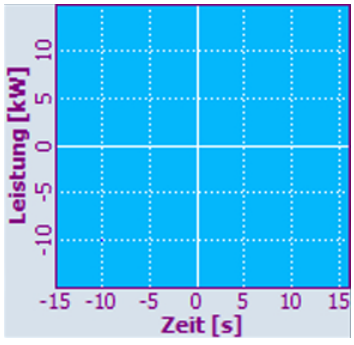



Figure 7-8 Scaling text outside the drawing area, vertical text alignment

## KeepAspectRatio – keep aspect ratio

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***KeepAspectRatio***")*<br>WriteCWProperty*(GraphVarName, "***KeepAspectRatio***", Value)* | |
| Description: | With this property, you define as to whether the view defined using setView() should be automatically aligned, adapted or extended, so that the aspect ratio from X to Y axis always remains the same. This property is especially relevant when displaying geometrical figures. For example, a circle or square is displayed as such, and does not become an ellipse or rectangle. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (bool) |
| | Return Value | Value to be set (bool): TRUE or FALSE |

## Example

```
setView(-15, -15, 15, 15)
setFillColor("#A0A0A4")
addCircle(0, 0, 5)
KeepAspectRatio = TRUE
```

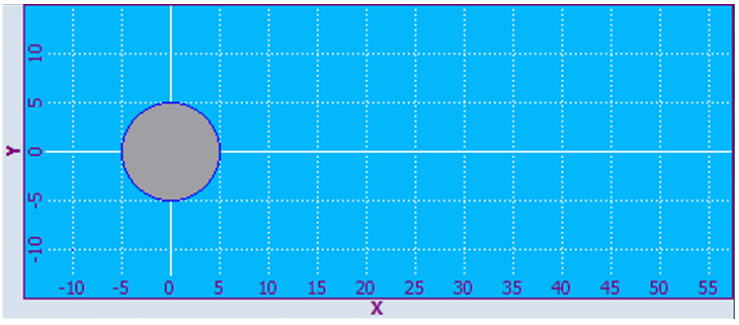



Figure 7-9 Aspect ratio from X to Y axis remain the same

```
KeepAspectRatio = TRUE
```

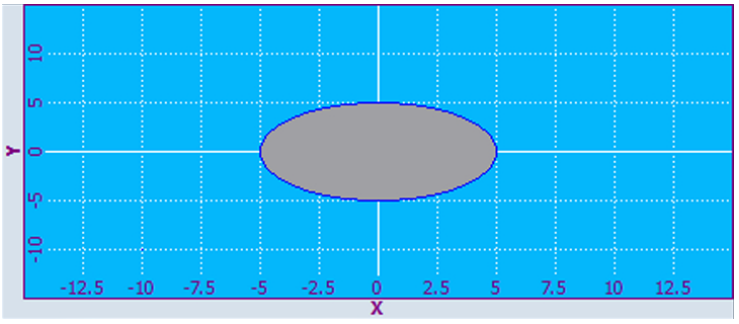



Figure 7-10 Aspect ratio from X to Y axis distorted

## SelectedContour – name of the currently selected contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***SelectedContour***")*<br>WriteCWProperty*(GraphVarName, "***SelectedContour***", Value)* | |
| Description: | Reads or sets the selection of the actual contour. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (QString) |
| | Value | Value to be set (QString) |

**Note**

In order to be able to execute operations on a contour, e.g. to add a graphic object, you must first select this.

## BackColor – background color

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***BackColor***")*<br>WriteCWProperty*(GraphVarName, "***BackColor***", Value)* | |
| Description: | The background color for labeling axes depends on the ScaleTextInsideChartArea property - also the scaling texts. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEsGraphCustomWidget |
| | Return Value | Read value of the property (QString) |
| | Value | Value to be set (QString) as RGB value in the form "#RRGGBB", for example, "#04B7FB" |

## ChartBackColor – background color of the drawing area

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, '***ChartBackColor***')*<br>WriteCWProperty*(GraphVarName, '***ChartBackColor***', Value)* | |
| Description: | Background color for the drawing area. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEsGraphCustomWidget |
| | Return Value | Read value of the property (QString) |
| | Value | Value to be set (QString) as RGB value in the form "#RRGGBB", for example, "#04B7FB" |

## ForeColor - foreground color for labeling and default pen color

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, '***ForeColor***')*<br>WriteCWProperty*(GraphVarName, '***ForeColor***', Value)* | |
| Description: | Foreground color for text and default pen color | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEsGraphCustomWidget |
| | Return Value | Read value of the property (QString) |
| | Value | Value to be set (QString) as RGB value in the form "#RRGGBB", for example, "#04B7FB" |

## ForeColorY2 - foreground color for labeling the second Y axis (right)

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, '***ForeColorY2***')*<br>WriteCWProperty*(GraphVarName, '***ForeColorY2***', Value)* | |
| Description: | Foreground color for the texts of the second Y axis (right) | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEsGraphCustomWidget |
| | Return Value | Read value of the property (QString) |
| | Value | Value to be set (QString) as RGB value in the form "#RRGGBB", for example, "#04B7FB" |

## GridColor - color of the grid lines

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "**GridColor**")*<br>WriteCWProperty*(GraphVarName, "**GridColor**", Value)* | |
| Description: | Color of the grid lines. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (QString) |
| | Value | Value to be set (QString) as RGB value in the form "#RRGGBB", for example, "#04B7FB" |

## GridColorY2 - color of the horizontal grid lines of the second Y axis (right)

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "**GridColorY2**")*<br>WriteCWProperty*(GraphVarName, "**GridColorY2**", Value)* | |
| Description: | Color of the horizontal grid lines of the second Y axis (right). | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (QString) |
| | Value | Value to be set (QString) as RGB value in the form "#RRGGBB", for example, "#04B7FB" |

## CursorColor - color of the cursor

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "**CursorColor**")*<br>WriteCWProperty*(GraphVarName, "**CursorColor**", Value)* | |
| Description: | Color of the cursor. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (QString) |
| | Value | Value to be set (QString) as RGB value in the form "#RRGGBB", for example, "#04B7FB" |

## ShowCursor - Display/hide cursor

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "**ShowCursor**")*<br>WriteCWProperty*(GraphVarName, "**ShowCursor**", Value)* | |
| Description: | Displays/hides cursor. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (bool) |
| | Value | Value to be set (bool): TRUE or FALSE |

**Note**

This function automatically refreshes the display.

## CursorX – cursor X position

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***CursorX***")*<br>WriteCWProperty*(GraphVarName, "***CursorX***", Value)* | |
| Description: | X position of the cursor. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (double) |
| | Value | Value to be set (double). |

**Note**

This function automatically refreshes the display.

## CursorY – cursor Y position

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***CursorY***")*<br>WriteCWProperty*(GraphVarName, "***CursorY***", Value)* | |
| Description: | Y position of the cursor. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (double) |
| | Value | Value to be set (double). |

**Note**

This function automatically refreshes the display.

## CursorY2 - cursor Y position referred to the second Y axis (right)

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***CursorY2***")*<br>WriteCWProperty*(GraphVarName, "***CursorY2***", Value)* | |
| Description: | Y position of the cursor referred to the second Y axis (right). | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (double) |
| | Value | Value to be set (double). |

**Note**

This function automatically refreshes the display.

### CursorStyle – cursor display type

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***CursorStyle***")*<br>WriteCWProperty*(GraphVarName, "***CursorStyle***", Value)* | |
| Description: | Cursor display type. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (int) |
| | Value | Value to be set (int):<br>0 (= cross)<br>1 (= crosshairs)<br>2 (= vertical line)<br>3 (= horizontal line)<br>4 (= horizontal and vertical line)) |

### ViewMoveZoomMode – response when zooming and shifting using gestures

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty*(GraphVarName, "***ViewMoveZoomMode***")*<br>WriteCWProperty*(GraphVarName, "***ViewMoveZoomMode***", Value)* | |
| Description: | The displayed view of the SIEsGraphCustomWidgets can be changed using gestures. This property allows you to define which type should be permitted. For example, in certain applications it may make sense that the view is only horizontally shifted and zoomed, e.g. when displaying measured value characteristics. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Read value of the property (int) |
| | Value | Value to be set (int):<br>0 (= off)<br>1 (= only horizontal)<br>2 (= only vertical)<br>3 (= horizontal and vertical) |

## 7.5.5 Functions

You can call the subsequently listed functions using the CallCWMethod() function.

### Example

Setting the view of the SIEsGraphCustomWidget link using display variable "MyGraphVar".

The result is written to register 0.

```
REG[0] = CallCWMethod("MyGraphVar", "setView", -8, -5, 11- 20)
```

## Overview

| Function | Description |
|---|---|
| setView | Sets coordinate system |
| setMaxContourObjects | Sets maximum number of objects for a contour (ring buffer) |
| addContour | Adds a contour |
| showContour | Shows a contour |
| hideContour | Hides a contour |
| hideAllContours | Hides all contours |
| removeContour | Removes a contour |
| clearContour | Deletes graphic object of a contour |
| fitViewToContours | Automatic adaptation of the view |
| fitViewToContour | Automatic adaptation of the view |
| findX | Searches in a contour |
| setPolylineMode | Displays points of a contour as polyline/curve |
| setIntegralFillMode | Displays points of a contour as polyline/curve |
| repaint, update | Refreshes view |
| addPoint | Adds a point to a contour |
| addLine | Adds a line to a contour |
| addRect | Adds a rectangle to a contour |
| addRoundedRect | Adds a rectangle with rounded corners to a contour |
| addEllipse | Adds an ellipse to a contour |
| addCircle | Adds a circle to a contour |
| addArc | Adds an arc to a contour |
| addText | Adds text to a contour |
| setPenWidth | Sets the pen width |
| setPenStyle | Sets the pen style |
| setPenColor | Sets the pen color |
| setFillColor | Sets fill color |
| setCursorPosition | Positions the cursor |
| setCursorPositionY2 | Positions the cursor referred to the second Y axis (right) |
| setCursorOnContour | Positions the cursor on contour |
| moveCursorOnContourBegin | Positions the cursor to the first graphic object of a contour |
| moveCursorOnContourEnd | Positions the cursor to the last graphic object of a contour |
| moveCursorOnContourNext | Positions the cursor to the next graphic object of a contour |
| serialize | Saves/restores the actual state |

## setView – sets the coordinate system

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "**setView**", x1, y1, x2, y2)* | |
| Description: | Using this function, you define the size of the coordinate system that should be displayed within the SIEsGraphCustomWidget. See also KeepAspectRatio property. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x1 | left-hand edge (double) |
| | y1 | upper edge (double) |
| | x2 | right-hand edge (double) |
| | y2 | lower edge (double) |

**Note**

This function automatically refreshes the display.

## Example

```
setView(-8, -5, 11, 20)
AxisNameX = "X"
AxisNameY = "Y"
ScaleTextOrientationYAxis = 1
ScaleTextEmbedded = false
KeepAspectRatio = false
update
```

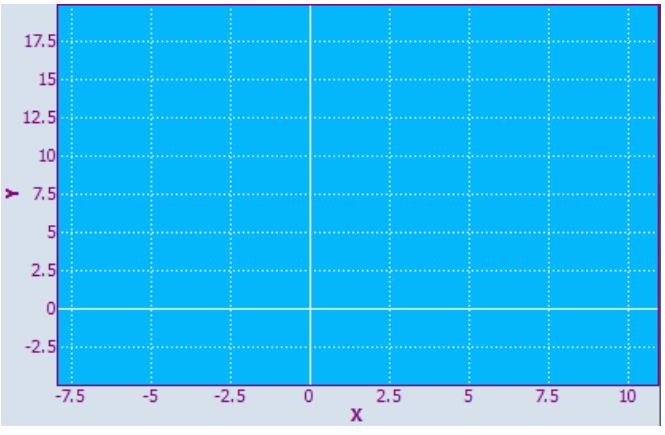



Figure 7-11 Example - setView

## setMaxContourObjects – sets the maximum number of objects of a contour (ring buffer)

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***setMaxContourObjects***", Value)*<br>ReturnValue = CallCWMethod*(GraphVarName, "***setMaxContourObjects***", Value, ContourName)* | |
| Description: | Using this function, you can define the size of a contour ring buffer. If the number of objects added to the contour exceeds this limit, then the oldest object is deleted. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | Value | Maximum number of graphic objects (uint) |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |

**Note**

This function automatically refreshes the display.

## addContour – add contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***addContour***", ContourName)*<br>ReturnValue = CallCWMethod*(GraphVarName, "***addContour***", ContourName, Selected)*<br>ReturnValue = CallCWMethod*(GraphVarName, "***addContour***", ContourName, Selected, AssignedY2)* | |
| Description: | Using this function, you can define a new contour.<br>Optionally, you can assign the contour to the coordinate system of the second Y axis (right). | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |
| | Selected | Contour should be selected (bool): TRUE or FALSE |
| | AssignedY2 | Contour should be assigned to the second Y axis (right)<br>(bool): TRUE or FALSE |

## showContour – shows a contour

| | |
|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***showContour***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***showContour***", ContourName)* |
| Description: | This function makes a contour visible. |

| | | |
|---|---|---|
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |

## hideContour – hide contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod(*GraphVarName, "***hideContour***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***hideContour***", ContourName)* | |
| Description: | This function hides a contour. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |

## showAllContours – shows all contours

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***showAllContours***")* | |
| Description: | This function makes all contours visible. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |

## hideAllContours – hide all contours

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***hideAllContours***")* | |
| Description: | This function hides all contours. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |

## removeContour – hides a contour

| | |
|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***removeContour***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***removeContour***", Contour-Name)* |
| Description: | Removes a contour. |

| | | |
|---|---|---|
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |

## clearContour – delete graphic object of a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***clearContour***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***clearContour***", ContourName)* | |
| Description: | Deletes all graphic objects on a contour. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |

## fitViewToContours – automatic adaptation of the view

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***fitViewToContours***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***fitViewToContours***", OnlyVisible)* | |
| Description: | Using this function, the view is automatically adapted to the contours. Depending on the transfer parameter, you define whether only all visible contours can be seen or all contours can be seen. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | OnlyVisible | Value to be set (bool): TRUE or FALSE |

## fitViewToContour – automatic adaptation of the view

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***fitViewToContour***", ContourName)* | |
| Description: | Using this function, the view is automatically adapted so that only a specific contours visible. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString) |

## findX – search in a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "**findX**", x)*<br>ReturnValue = CallCWMethod*(GraphVarName, "**findX**", x, ContourName)*<br>ReturnValue = CallCWMethod(*GraphVarName, "**findX**", x, ContourName, SetCursor)* | |
| Description: | Using this function, you can search for a previously defined point with a specific X coordinate in a contour. You obtain the Y coordinate as result. Optionally, you can also simultaneously position the cursor to this point. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (QString): If successful, the associated Y value is supplied |
| | x | X value to be searched for (double) |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour |
| | SetCursor | Set cursor (bool): TRUE or FALSE |

**Note**

The display is automatically updated if you set the cursor using this function.

## setPolylineMode – displays points on the contour as polyline/curve

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "**setPolylineMode**", SetPoly)* | |
| Description: | All of the points of the actual contour added after this function call are connected to create a polyline/curve. This function is contour-specific, and can be activated and deactivated on a segment-for-segment basis. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | SetPoly | PolylineMode (bool): TRUE = on |

## Example

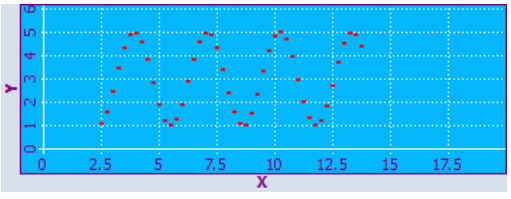



Figure 7-12 Example - PolylineMode: Off

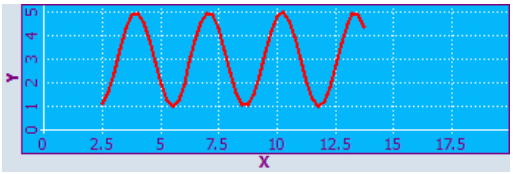



Figure 7-13 Example - PolylineMode: To

## setIntegralFillMode – fills the areas between points, lines and arcs and the X axis

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "**setIntegralFillMode**", SetIntFill)* | |
| Description: | Areas between points, lines and arcs of the X axis are filled with the current fill color (independent of whether the points are +Y or –Y).<br>This function is contour-specific, and can be activated and deactivated on a segment-for-segment basis. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | SetIntFill | IntegralFillMode (bool): TRUE = on |

## Example

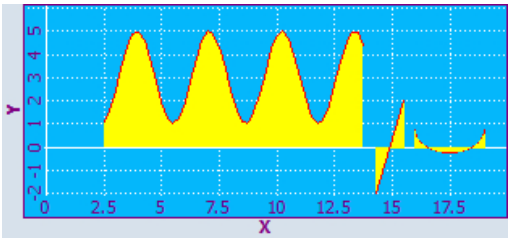



Figure 7-14 Example - setIntegralFillMode: on

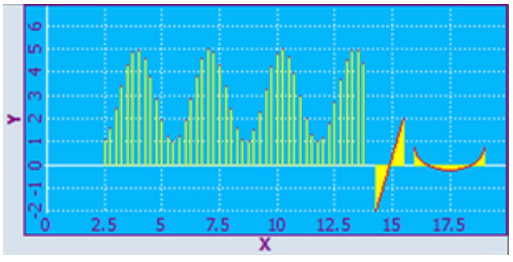



Figure 7-15 Example - setIntegralFillMode: Off

## repaint, update – update view

| | |
|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "**repaint**")*<br>ReturnValue = CallCWMethod*(GraphVarName, "**update**")* |
| Description: | Using this function, you can manually refresh the display. |

| | | |
|---|---|---|
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |

- update()
  The function refreshes the view, assuming that the Widget refresh is not deactivated, and the Widget is not hidden. The function is optimized so that the application performance is maintained, and the view does not flicker as a result of excessively high refresh rates.
- repaint()
  The function refreshes the view directly after the function call. Therefore, you should only use the repaint function if an immediate refresh is absolutely necessary, e.g. for animation purposes

It is recommended that you always work with the update() function. Internally, the SIEsGraphCustomWidget always works with the function update() e.g. for setView().

## addPoint – add point to a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "**addPoint**", x, y)*<br>ReturnValue = CallCWMethod*(GraphVarName, "**addPoint**", x, y, DummyPoint)* | |
| Description: | Adds a point to the currently selected contour Further, you can set a dummy point; although it does not show up in the drawing, it can be positioned with the cursor. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x | x coordinate (double) |
| | y | y coordinate (double) |
| | DummyPoint | Point is treated as invisible (bool): TRUE = yes |

## addLine – add line to a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "**addLine**", x1, y1, x2, y2)* | |
| Description: | Adds a line to the currently selected contour | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x1 | x coordinate of the starting point (double) |
| | y1 | y coordinate of the starting point (double) |
| | x2 | x coordinate of the end point (double) |
| | y2 | y coordinate of the end point (double) |

## addRect – add rectangle to a contour

| | |
|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "**addRect**", x1, y1, x2, y2)* |
| Description: | Adds a rectangle to the currently selected contour. |

| | | |
|---|---|---|
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x1 | left-hand edge (double) |
| | y1 | left-hand edge (double) |
| | x2 | right-hand edge (double) |
| | y2 | lower edge (double) |

## addRoundedRect – add rectangle with rounded corners to a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***addRoundedRect***", x1, y1, x2, y2, r)* | |
| Description: | Adds a rectangle with rounded off corners to the currently selected contour. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x1 | left-hand edge (double) |
| | y1 | upper edge (double) |
| | x2 | upper edge (double) |
| | y2 | upper edge (double) |
| | r | Radius (double) |

## addElipse – add ellipse to a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***addEllipse***", x1, y1, x2, y2, radius)* | |
| Description: | Adds an ellipse to the currently selected contour. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x1 | left-hand edge (double) |
| | y1 | upper edge (double) |
| | x2 | right-hand edge (double) |
| | y2 | lower edge (double) |
| | Radius | Radius (double) |

## addCircle – add circle to a contour

| | |
|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***addCircle***", x, y, radius)* |
| Description: | Adds a circle to the currently selected contour |

| | | |
|---|---|---|
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x | left-hand edge (double) |
| | y | upper edge (double) |
| | Radius | Radius (double) |

## addArc – add arc to a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "addArc", x1, y1, x2, y2, StartAngle, SpanAngle* | |
| Description: | Adds an arc to the currently selected contour | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x1 | left-hand edge (double) |
| | y1 | upper edge (double) |
| | x2 | right-hand edge (double) |
| | y2 | lower edge (double) |
| | StartAngle | Starting angle in degrees (double) |
| | SpanAngle | Opening angle in degrees (double) |

## addText – add text to a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "addText", x, y, Text)* | |
| Description: | Adds a text to the currently selected contour | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x | left-hand edge (double) |
| | y | upper edge (double) |
| | Text | Text (QString) |

## setPenWidth – sets the pen width

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "setPenWidth")* ReturnValue = CallCWMethod*(GraphVarName, "setPenWidth", width)* | |
| Description: | Defines the pen width from the instant that the function was called for the following objects that were added to the actual contour. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | width | Pen width (double).<br>If a pen width is not specified, then the default pen width is set. |

## setPenStyle – sets the pen style

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***setPenStyle***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***setPenStyle***", style)* | |
| Description: | Defines pen style from the instant that the function is called for the following objects that are added to the actual contour. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | style | 1 : Solid line (___)<br>2 : Interrupted line (_ _)<br>3 : Dotted line (...)<br>4 : Line-point-line (_._)<br>5 : Line-point-point (_..) |

## setPenColor – sets the pen color

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***setPenColor***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***setPenColor***", Color)* | |
| Description: | Defines the pen color from the instant that the function was called for the following objects that were added to the actual contour. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | Color | Value to be set (QString) as RGB value in the form "#RRGGBB", for example, "#04B7FB" |

## setFillColor – set fill color

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***setFillColor***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***setFillColor***", Color)* | |
| Description: | Defines the fill color from the instant that the function was called for the following objects that were added to the actual contour. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | Color | Value to be set (QString) as RGB value in the form "#RRGGBB", for example, "#04B7FB" |

## setCursorPosition – position cursor

| | |
|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***setCursorPosition***", x, y)* |
| Description: | Sets the cursor to the specified position. |

| | | |
|---|---|---|
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x | x coordinate (double) |
| | y | y coordinate (double) |

**Note**

This function automatically refreshes the display.

## setCursorPositionY2Cursor – position cursor referred to the second Y axis (right)

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***setCursorPositionY2***", x, y2)* | |
| Description: | Sets the cursor to the specified position referred to the second Y axis (right) | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | x | x coordinate (double) |
| | y2 | Y coordinate referred to the second Y axis (right) (double) |

**Note**

This function automatically refreshes the display.

## setCursorOnContour – position cursor on contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***setCursorOnContour***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***setCursorOnContour***", ContourName)* | |
| Description: | Sets the cursor to the last cursor position within a contour. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |

**Note**

This function automatically refreshes the display.

## Example

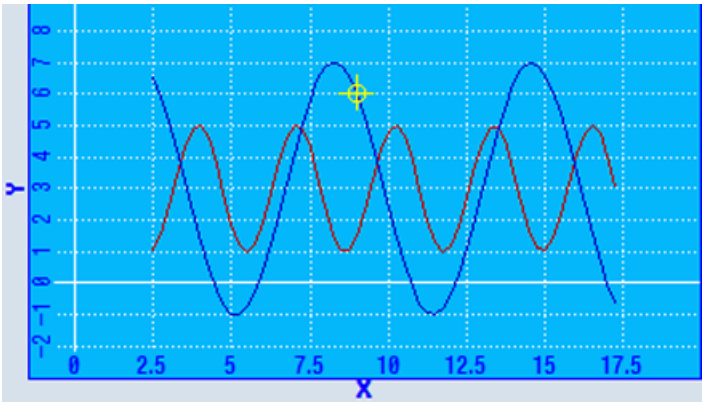



Figure 7-16 Example - setCursorOnContour

## moveCursorOnContourBegin – position the cursor to the first graphic object of a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***moveCursorOnContourBegin***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***moveCursorOnContourBegin***", ContourName)* | |
| Description: | Starting from the actual cursor position within a contour, using this function you can navigate the cursor to the first point object of a contour. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |

**Note**

This function automatically refreshes the display.

## moveCursorOnContourEnd – position the cursor to the last graphic object of a contour

| | |
|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***moveCursorOnContourEnd***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***moveCursorOnContourEnd***", ContourName)* |
| Description: | Starting from the actual cursor position within a contour, using this function you can navigate the cursor to the last point object of a contour. |

| | | |
|---|---|---|
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |

**Note**

This function automatically refreshes the display.

## moveCursorOnContourNext – position the cursor to the next graphic object of a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***moveCursorOnContourNext***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***moveCursorOnContourNext***", ContourName)* | |
| Description: | Starting from the actual cursor position within a contour, using this function you can navigate the cursor to the next point object of a contour. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |

**Note**

This function automatically refreshes the display.

## moveCursorOnContourBack – position the cursor to the previous graphic object of a contour

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***moveCursorOnContourBack***")*<br>ReturnValue = CallCWMethod*(GraphVarName, "***moveCursorOnContourBack***", ContourName)* | |
| Description: | Starting from the actual cursor position within a contour, using this function you can navigate the cursor to the previous point object of a contour. | |
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | ContourName | Name of the contour (QString).<br>If a contour is not specified, then the call automatically refers to the currently selected contour. |

**Note**

This function automatically refreshes the display.

### serialize – save/restore the actual state

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod*(GraphVarName, "***serialize***", FilePath, IsStoring)* | |
| Description: | Using this function, when required, you can write the actual state and content of the SIEsGraphWidget in binary form to a file or datastream and also restore it. | |
| Parameter: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | Return Value | Error code (bool): TRUE = successful |
| | FilePath | Complete path mit file name (QString) |
| | IsStoring | Save/restore (bool): TRUE = save |

**Note**

This function automatically refreshes the display.

## 7.5.6 Signals

You can capture the subsequently described ViewChanged signal in the configuration and respond appropriately.

### Overview

| Function | Description |
|---|---|
| ViewChanged | Changing the view |

### ViewChanged – changing the view

| | |
|---|---|
| Syntax: | SUB*(on_<GraphVarName>*_**ViewChanged***)*<br>…<br>END_SUB |
| Description: | The "ViewChanged“ signal is sent if the view changes. You can respond to this in the configuration in a specific SUB method. The SIGARG parameters are appropriately set. |

| | | |
|---|---|---|
| Parameters: | GraphVarName | Name of the display variable which contains a SIEs-GraphCustomWidget |
| | SIGARG[0] | left-hand edge (double) |
| | SIGARG[1] | upper edge (double) |
| | SIGARG[2] | right-hand edge (double) |
| | SIGARG[3] | lower edge (double) |

## Example

```
DEF MyGraphVar = (W///,"slesgraphcustomwidget.SlEsGraphCustomWidget"/////
10,10,340,340/0,0,0,0)
```

```
SUB(on_MyGraphVar_ViewChanged
      DLGL("Current view: " << SIGARG[0] << ", " << SIGARG[1] << ", " <<   SIGARG[2]
<< ", " << SIGARG[3]
END_SUB
```

# 7.6 SlEsTouchButton

## 7.6.1 SlEsTouchButton

### General information

You can create even functionally demanding applications simply by using Run MyScreens. For touch operation you can configure buttons that can be freely located (TouchButtons). The following attributes apply when configuring TouchButtons:

- The two possible state changes of the "clicked" and "checked" buttons can be queried in the configuration and the appropriate actions initiated.
- TouchButtons can be operated with single or multitouch, mouse and keyboard.
- The TouchButton is displayed corresponding to the actual resolution. This also applies to the font and displayed graphics.
- The TouchButton has two styles "Softkey look&feel" and "Application-specific". In the "Softkey look&feel" style, the TouchButton is displayed corresponding to the Operate softkeys. Both display styles have functions, for example, to scale graphics.
- TouchButtons are implemented and made available as custom widgets.

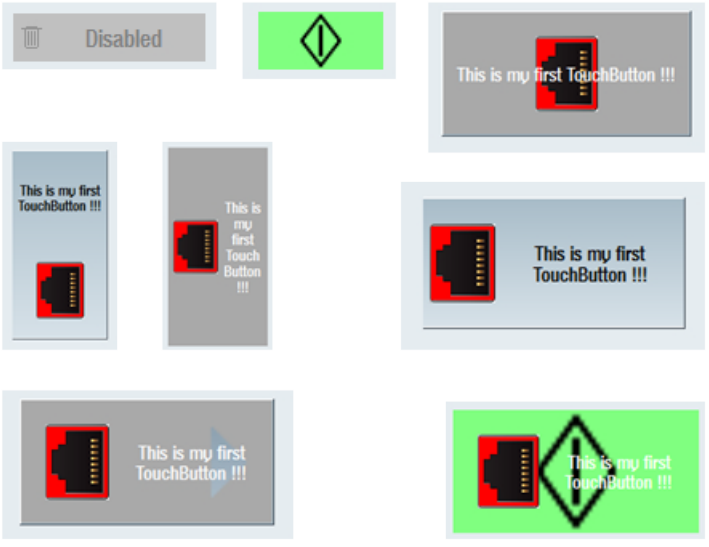



Figure 7-17 Examples of TouchButtons

### Example

```
//M(MyTBMask/"My CustomWidget TouchButton ...")
DEF MyTB1 = (W///,"slesstdcw.SlEsTouchButton"/////70,20,200,100/0,0,0,0)
LOAD
   WRITECWPROPERTY("MyTB1", "text", "This is my first TouchButton !!!")
```

```
//M(MyTBMask/"My CustomWidget TouchButton ...")
   WRITECWPROPERTY("MyTB1", "textPressed", "This is my first TouchButton (pressed)!!!")
   WRITECWPROPERTY("MyTB1", "picture", "dsm_remove_trashcan_red.png") WRITECWPROPERTY("MyTB1", "pic-
tureAlignment", "left")
   WRITECWPROPERTY("MyTB1", "scalePicture", FALSE)
   WRITECWPROPERTY("MyTB1", "picturePressed", "slsu_topology_empty_round_slot.png") WRITECWPROPER-
TY("MyTB1", "picture", "slsu_topology_empty_slot_left_error.png")
END_LOAD
```

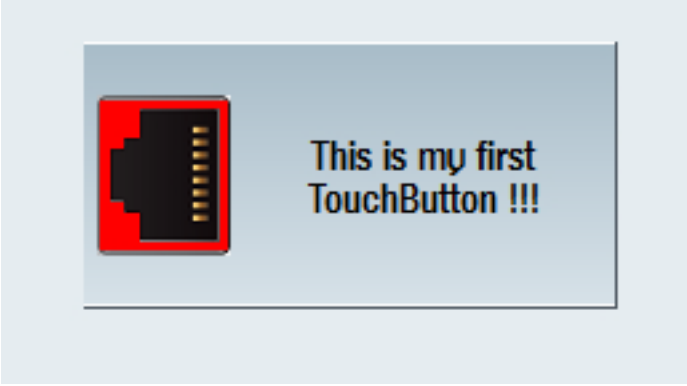



Figure 7-18 Example, "This is my first TouchButton !!!"

## 7.6.2 Reading and writing properties

### Description

The properties listed in the following section are read with ReadCWProperty() and written with WriteCWProperty().

### Examples

Reading the "Text" property of the SIEsTouchButton linked using display variable "MyTouchButton". The result is written to register 0.

```
REG[0] = ReadCWProperty("MyTouchButton", "Text")
```

Writing value "sk_ok.png" in the "Picture" property of the SIEsTouchButton, linked using display variable "MyTouchButton".

```
WriteCWProperty("MyTouchButton", "Picture", "sk_ok.png")
```

## 7.6.3 Properties

### Overview

| Property | Description |
|---|---|
| ButtonStyle | TouchButton display style |
| Flat | Displayed flat or in 3D |
| Enabled | Activate/deactivate operability |
| Checkable | Activate/deactivate toggle function |
| Checked | TouchButton is presently checked |
| ShowFocusRect | Display focus square if the TouchButton has the input focus |
| Picture | Picture (image) that should be displayed if the TouchButton is in the normal, non-actuated state |
| PicturePressed | Picture (image) to be displayed if the TouchButton is pressed or checked |
| PictureAlignment<br>PictureAlignmentString | Alignment of the picture |
| ScalePicture | Picture is scaled (stretched or compressed) |
| PictureKeepAspectRatio | Picture aspect ratio |
| Text | Normal display text |
| TextPressed | Text that is displayed if the TouchButton is pressed |
| textAlignment<br>textAlignmentString | Alignment of the text |
| TextAlignedToPicture | Text is aligned relative to the picture - activate/deactivate |
| BackgroundPicture | Background screen to be displayed |
| BackgroundPictureAlignment<br>BackgroundPictureAlignmentString | Alignment of the background picture |
| ScaleBackgroundPicture | Background picture is scaled (stretched or compressed) |
| BackgroundPictureKeepAspectRatio | Picture aspect ratio |
| BackColor | Background color if the TouchButton is in the normal and non-actuated state |
| BackColorChecked | Background color if the TouchButton is checked |
| BackColorPressed | Background color if the TouchButton is presently pressed |
| BackColorDisabled | Background color if the TouchButton cannot be operated |
| TextColor | Text color if the TouchButton is in the normal and non-actuated state |
| TextColorChecked | Text color if the TouchButton is checked |
| TextColorPressed | Text color if the TouchButton is presently pressed |
| TextColorDisabled | Text color if the TouchButton cannot be operated |

## Button style

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**ButtonStyle**")<br>WriteCWProperty(*TouchButtonVarName*, "**ButtonStyle**", *Value*) | |
| Description: | Reads/sets the TouchButton style | |
| Parameter: | TouchButtonVarName | Name of the display variable that contains a SIEs-TouchButton |
| | Return Value | read property value (int) |
| | Value | value to be set (int)<br>0 = Softkey Look&Feel (default)<br>1 = user-specific |

With the setting "0 = Softkey Look&Feel", the TouchButton is displayed and responds just like a softkey. The currently set skin is taken into account – see display machine data 9112 = $MM_HMI_SKIN.

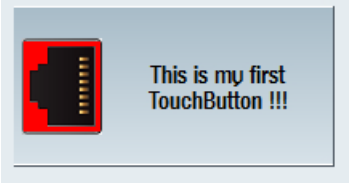



Figure 7-19 ButtonStyle - Softkey Look&Feel

With setting "1 = user-specific", the text and background color can be defined according to the TouchButton state. Further, there is a 3D effect, which allows pictures to be automatically scaled and influences the distances from the edges.

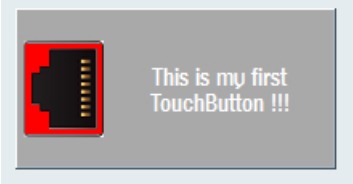



Figure 7-20 ButtonStyle - user-specific

## Flat – display type

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**Flat**")<br>WriteCWProperty(*TouchButtonVarName*, "**Flat**", *Value*) | |
| Description: | Reads/sets as to whether the TouchButton is shown flat (default) or in 3D<br>**Note:** This property is only available with active style (ButtonStyle) "1= user-specific". | |
| | TouchButtonVarName | Name of the display variable that contains a SIEs-TouchButton |
| Parameter: | Return Value | read value of the property (bool) |
| | Value | value to be set (bool): TRUE (default) or FALSE |

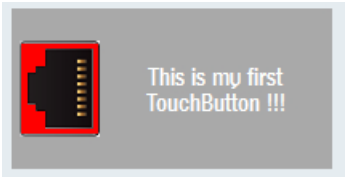



Figure 7-21 Flat display style

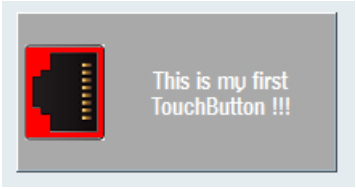



Figure 7-22 3D display style

## Enabled – operability of the TouchButton

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty( *TouchButtonVarName*, "**Enabled**")<br>WriteCWProperty( *TouchButtonVarName*, "**Enabled**", *Value*) | |
| Description: | Read/sets as to whether the TouchButton can be operated (= TRUE) - or cannot be operated (FALSE). | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read value of the property (bool) |
| | Value | Value to be set (bool): TRUE (default) or FALSE |

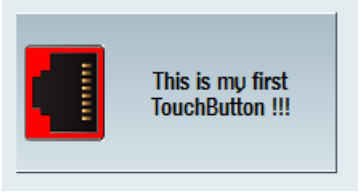



Figure 7-23 TouchButton can be operated

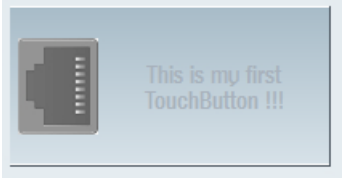



Figure 7-24 TouchButton cannot be operated

**Note**

If a TouchButton that cannot be operated is clicked, then signal "clickedDisabled" is sent. The signal can be evaluated, for example an appropriate message issued as to why the TouchButton and the associated function presently cannot be operated. In the deactivated state, the picture or background picture is automatically displayed in shades of gray.

## Checkable – activate/deactivate toggle function

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**Checkable**")<br>WriteCWProperty(*TouchButtonVarName*, "**Checkable**", *Value*) | |
| Description: | Reads/sets the TouchButton toggle function. | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | read value of the property (bool) |
| | Value | value to be set (bool): TRUE or FALSE (default) |

**Note**

In the default setting, after it has been released, the TouchButton returns to its normal state (the same as a pushbutton) However, if the TouchButton toggle functionality is activated (TRUE), then after it is press, it initially remains in the pressed position. If it is actuated again, then it changes back into its normal state (same as a switch).

See also the "Checked" property.

## Checked – actual toggle state

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**Checked**")<br>WriteCWProperty(*TouchButtonVarName*, "**Checked**", *Value*) | |
| Description: | Reads/sets the actual toggle state of the TouchButton | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | read value of the property (bool) |
| | Value | value to be set (bool): TRUE or FALSE (default) |

If the toggle function is activated as the property "Checkable" is "TRUE", then the actual toggle state can be read or set using the "Checked" property.

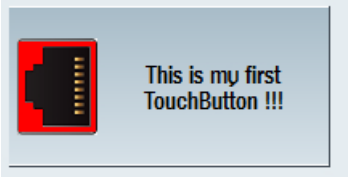



Figure 7-25 Unchecked

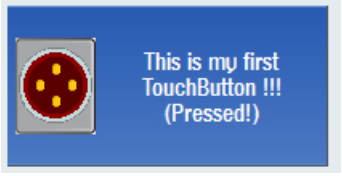



Figure 7-26 Checked

For both states, text and picture can be specifically displayed.

**Note**

See also the "Checkable" property.

## ShowFocusRect – focus rectangle display

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**ShowFocusRect**")<br>WriteCWProperty(*TouchButtonVarName*, "**ShowFocusRect**", *Value*) | |
| Description: | Reads/sets as to whether the TouchButton should display a focus rectangle as soon as it is assigned the input focus. | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SlEsTouch-Button |
| | Return Value | Read value of the property (bool) |
| | Value | Value to be set (bool): TRUE or FALSE (default) |

The focus frame color is automatically set with the Operate color setting, in the following example "orange".

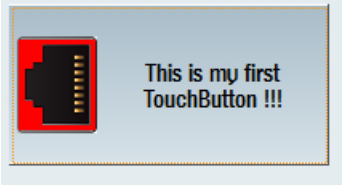



Figure 7-27 ShowFocusRect

## Picture – displayed picture

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**Picture**")<br>WriteCWProperty(*TouchButtonVarName*, "**Picture**", *Value*) | |
| Description: | Reads/sets the picture to be displayed if the TouchButton is in its quiescent state (not actuated). | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SlEsTouch-Button |
| | Return value | read value of the property (string) |
| | Value | value to be set (string) |

The file name is specified as value, for example "sk_ok.png". The TouchButton automatically determines the correct picture file from the current resolution directory (see Chapter Using display images/graphics (Page 61)).

In the deactivated state, the picture is automatically displayed in shades of gray.

**Note**

See also properties - "Text", "TextAlignedToPicture", "PictureAlignment", "ScalePicture", "PictureKeepAspectRatio".

## PicturePressed – picture in the pressed state

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**PicturePressed**")<br>WriteCWProperty(*TouchButtonVarName*, "**PicturePressed**", *Value*) | |
| Description: | Reads/sets the picture to be displayed if the TouchButton is pressed or checked. | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return value | read value of the property (string) |
| | Value | value to be set (string) |

If a value is not specified (empty string " "), then in this state, the TouchButton shows the picture specified with the "Picture" property.

The file name is specified as value, for example "sk_ok.png". The TouchButton automatically determines the correct picture file from the current resolution directory (see Chapter Using display images/graphics (Page 61)).

In the deactivated state, the picture is automatically displayed in shades of gray.

**Note**

See also properties - "Text", "TextAlignedToPicture", "PictureAlignment", "ScalePicture", "PictureKeepAspectRatio".

## PictureAlignment – picture alignment

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**PictureAlignment**")<br>WriteCWProperty(*TouchButtonVarName*, "**PictureAlignment**", *Value*) | |
| Description: | Reads/sets the alignment of the picture on the TouchButton. | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read value of the property (int) |
| | Value | Value to be set (int):<br>1 = left (default)<br>2 = right<br>32 = top<br>64 = bottom<br>128 = centered |

**Note**

See Chapter Positioning and aligning picture and text (Page 269).
Also see properties - "Text", "TextAlignedToPicture", "PictureAlignmentString", "ScalePicture", "PictureKeepAspectRatio".

## PictureAlignmentString – picture alignment

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**PictureAlignment-String**")<br>WriteCWProperty(*TouchButtonVarName*, "**PictureAlignmentString**", *Value*) | |
| Description: | Reads/sets the alignment of the picture. This property is analogous to the "PictureAlignment" property. However, the value is not transfigured here as number (int), but as string. | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return value | read value of the property (string) |
| | Value | value to be set (string):<br>"Left" = left (default)<br>"Right" = right<br>"Top" = top<br>"Bottom" = bottom<br>"Center" = centered |

**Note**

See Chapter Positioning and aligning picture and text (Page 269).

See also properties - "Text", "TextAlignedToPicture", "PictureAlignment", "ScalePicture", "PictureKeepAspectRatio".

## ScalePicture – scale picture

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**ScalePicture**")<br>WriteCWProperty(*TouchButtonVarName*, "**ScalePicture**", *Value*) | |
| Description: | Reads/sets whether the TouchButton (depending on the alignment) should scale the picture to be displayed. | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read value of the property (bool) |
| | Value | value to be set (bool): TRUE or FALSE (default) |

**Note**

See Chapter Positioning and aligning picture and text (Page 269).

### PictureKeepAspectRatio – scale picture - keeping the aspect ratio

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**PictureKeepAspectRatio**")<br>WriteCWProperty(*TouchButtonVarName*, "**PictureKeepAspectRatio**", *Value*) | |
| Description: | Reads/sets, whether, when scaling (Property "ScalePicture" = "TRUE"), the aspect ratio of the picture should be kept or not. | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | read value of the property (bool) |
| | Value | Value to be set (bool): TRUE (default) or FALSE |

**Note**

See Chapter Positioning and aligning picture and text (Page 269).

### Text – display text

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**Text**")<br>WriteCWProperty(*TouchButtonVarName*, "**Text**", *Value*) | |
| Description: | Reads/sets the displayed text if the TouchButton is in its quiescent state (not actuated). | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return value | read value of the property (string) |
| | Value | value to be set (string) |

The texts are automatically wrapped around at the word limits in the rectangle provided.

Line breaks are only forced as a result of the system if the text to be displayed comes from a language file.

**Note**

See Chapter Language-dependent texts (Page 271).

### TextPressed – displayed text in the pressed state

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**TextPressed**")<br>WriteCWProperty(*TouchButtonVarName*, "**TextPressed**", *Value*) | |
| Description: | Reads/sets the displayed text if the TouchButton is pressed or checked. | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return value | read value of the property (string) |
| | Value | value to be set (string) |

The texts are automatically wrapped around at the word limits in the rectangle provided.

Line breaks are only forced as a result of the system if the text to be displayed comes from a language file.

**Note**

See Chapter Language-dependent texts (Page 271).

## TextAlignment – alignment of the text

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**TextAlignment**")<br>WriteCWProperty(*TouchButtonVarName*, "**TextAlignment**", *Value*) | |
| Description: | Reads/sets the alignment of the text on the TouchButton | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | read property value (int) |
| | Value | Value to be set (int):<br>1 = left<br>2 = right<br>32 = top<br>64 = bottom<br>128 = centered (default)<br>(see examples below) |

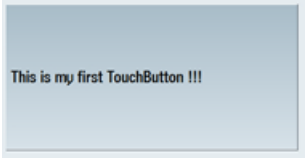



Figure 7-28 TextAlignment - left




Figure 7-29 TextAlignment - right

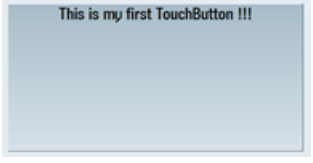



Figure 7-30 TextAlignment - top

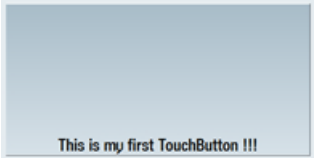



Figure 7-31 TextAlignment - bottom

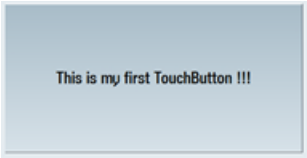



Figure 7-32 TextAlignment - centered

**Note**

See Chapter Positioning and aligning picture and text (Page 269).

See also properties "Text", "TextAlignmentString", "TextAlignedToPicture".

## TextAlignmentString – alignment of the text

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**TextAlignmentString**")<br>WriteCWProperty(*TouchButtonVarName*, "**TextAlignmentString**", *Value*) | |
| Description: | Reads/sets the alignment of the text. This property is analogous to the "TextAlignment" property. The value is not transfigured here as number (int), but as string. | |
| Parameter: | TouchButtonVarName | Name of the display variable that contains a SIEsTouchButton |
| | Return Value | read value of the property (string) |
| | Value | value to be set (string):<br>"Left" = left<br>"Right" = right<br>"Top" = top<br>"Bottom" = bottom<br>"Center" = centered (default) |

**Note**

See Chapter Positioning and aligning picture and text (Page 269).

See also properties - "Text", "TextAlignment", "TextAlignedToPicture".

## TextAlignedToPicture – text aligned relative to the picture

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "TextAlignedToPicture")<br>WriteCWProperty(*TouchButtonVarName*, "**TextAlignedToPicture**", *Value*) | |
| Description: | Reads/sets whether the displayed text should be positioned relative to the picture. If FALSE is set here, then the text is shown centered on the TouchButton. | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | read value of the property (bool) |
| | Value | value to be set (bool): TRUE (default) or FALSE |

**Note**

See Chapter Positioning and aligning picture and text (Page 269).
See also properties "Text", "TextPressed", "Picture", "PicturePressed".

## BackColor – background color

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, " **BackColor**")<br>WriteCWProperty(*TouchButtonVarName*, " **BackColor**", *Value*) | |
| Description: | Reads/sets the background color if the TouchButton is in its quiescent state (not actuated). | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read color of the property (string) |
| | Value | Value to be set (string) as RGB value in the form "#RRGGBB",<br>for example, "#04B7FB" |

**Note**

This property is only available with active style "1= user-specific".

## BackColorChecked – background color in the checked state

| | |
|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**BackColorChecked**")<br>WriteCWProperty(*TouchButtonVarName*, " **BackColorChecked**", *Value*) |
| Description: | Reads/sets the background color if the TouchButton is in the clicked state (toggle function). See properties "Checked“, "Checkable“ |

| | | |
|---|---|---|
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read color of the property (string) |
| | Value | Value to be set (string) as RGB value in the form "#RRGGBB",<br>for example, "#04B7FB" |

**Note**

This property is only available with active style "1= user-specific".

**Note**

See also properties "Checked", "Checkable".

## BackColorPressed – background color in the pressed state

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**BackColorPressed**")<br>WriteCWProperty(*TouchButtonVarName*, " **BackColorPressed**", *Value*) | |
| Description: | Reads/sets the background color if the TouchButton is in the pressed state | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read color of the property (string) |
| | Value | Value to be set (string) as RGB value in the form "#RRGGBB"<br>e.g. "#04B7FB" |

**Note**

This property is only available with active style "1= user-specific".

## BackColorDisabled – background color in the deactivated state

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**BackColorDisabled**")<br>WriteCWProperty(*TouchButtonVarName*, " **BackColorDisabled**", *Value*) | |
| Description: | Reads/sets the background color if the TouchButton is in the deactivated state. | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read color of the property (string) |
| | Value | Value to be set (string) as RGB value in the form "#RRGGBB",<br>for example, "#04B7FB" |

**Note**

This property is only available with active style "1= user-specific".

**Note**

See also the " Enabled" property.

## TextColor – text color

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**TextColor**")<br>WriteCWProperty(*TouchButtonVarName*, " **TextColor**", *Value*) | |
| Description: | Reads/sets the text color if the TouchButton is in its quiescent state (not actuated). | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read color of the property (string) |
| | Value | Value to be set (string) as RGB value in the form "#RRGGBB",<br>for example, "#04B7FB" |

**Note**

This property is only available with active style "1= user-specific".

## TextColorChecked – text color in the checked state

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**TextColorChecked**")<br>WriteCWProperty(*TouchButtonVarName*, " **TextColorChecked**", *Value*) | |
| Description: | Reads/sets the text color if the TouchButton is in the checked state (toggle function). | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read color of the property (string) |
| | Value | Value to be set (string) as RGB value in the form "#RRGGBB",<br>for example, "#04B7FB" |

**Note**

This property is only available with active style "1= user-specific".

**Note**

See also properties "Checked", "Checkable".

## TextColorPressed – text color in the pressed state

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**TextColorPressed**")<br>WriteCWProperty(*TouchButtonVarName*, "**TextColorPressed**", *Value*) | |
| Description: | Reads/sets the text color if the TouchButton is in the pressed state | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read color of the property (string) |
| | Value | Value to be set (string) as RGB value in the form "#RRGGBB",<br>for example, "#04B7FB" |

**Note**

This property is only available with active style "1= user-specific".

## TextColorDisabled – text color in the deactivated state

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**TextColorDisabled**")<br>WriteCWProperty(*TouchButtonVarName*, "**TextColorDisabled**", *Value*) | |
| Description: | Reads/sets the text color if the TouchButton is in the deactivated state | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read color of the property (string) |
| | Value | Value to be set (string) as RGB value in the form "#RRGGBB",<br>for example, "#04B7FB" |

**Note**

This property is only available with active style "1= user-specific".

**Note**

See also the " Enabled" property.

## BackgroundPicture – background picture

| | |
|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**BackgroundPic-ture**")<br>WriteCWProperty(*TouchButtonVarName*, "**BackgroundPicture**", *Value*) |
| Description: | Reads/sets the permanent background picture to be displayed, i.e. independent of the state. |

| | | |
|---|---|---|
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | read value of the property (string) |
| | Value | value to be set (string) |

The file name is specified as value, for example "sk_ok.png". The TouchButton automatically determines the correct picture file from the current resolution directory (see Chapter Using display images/graphics (Page 61)).

In the deactivated state, the background picture is automatically shown in shades of gray.

**Note**

See Chapter Using display images/graphics (Page 61).

See also properties - "BackgroundPictureAlignment", "ScaleBackgroundPicture", "BackgroundPictureKeepAspectRatio".

## BackgroundPictureAlignment – alignment of the background picture

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**BackgroundPictur-eAlignment**")<br>WriteCWProperty(*TouchButtonVarName*, "**BackgroundPictureAlignment**", *Val-ue*) | |
| Description: | Reads/sets the alignment of the background picture. | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Read value of the property (int) |
| | Value | Value to be set (int):<br>1 = left<br>2 = right<br>32 = top<br>64 = bottom<br>128 = centered (default) |

**Note**

See Chapter Positioning and aligning picture and text (Page 269).

See also properties - "BackgroundPictureAlignment", "ScaleBackgroundPicture", "BackgroundPictureKeepAspectRatio".

## BackgroundPictureAlignmentString – alignment of the background picture

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**BackgroundPictureAlignmentString**")<br>WriteCWProperty(*TouchButtonVarName*, "**BackgroundPictureAlignmentString**", *Value*) | |
| Description: | Reads/sets the alignment of the background picture. This property is analogous to the property "BackgroundPictureAlignment". However, the value is not transfigured here as number (int), but as string. | |
| Parameter: | TouchButtonVarName | Name of the display variable that contains a SIEsTouchButton |
| | Return Value | Read value of the property (int) |
| | Value | Value to be set (int):<br>"Left" = left<br>"Right" = right<br>"Top" = top<br>"Bottom" = bottom<br>"Center" = centered (default) |

**Note**

See Chapter Positioning and aligning picture and text (Page 269).

See also properties - "BackgroundPictureAlignment", "ScaleBackgroundPicture", "BackgroundPictureKeepAspectRatio".

## ScaleBackgroundPicture – scale background picture

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**ScalePicture**")<br>WriteCWProperty(*TouchButtonVarName*, "**ScalePicture**", *Value*) | |
| Description: | Reads/sets whether the TouchButton (depending on the alignment) should scale the picture to be displayed. | |
| Parameter: | TouchButtonVarName | Name of the display variable that contains a SIEsTouchButton |
| | Return value | read value of the property (bool) |
| | Value | value to be set (bool): TRUE or FALSE (default) |

**Note**

See Chapter Positioning and aligning picture and text (Page 269).

See also properties - "BackgroundPictureAlignment", "ScaleBackgroundPicture", "BackgroundPictureKeepAspectRatio".

### BackgroundPictureKeepAspectRatio – scale background picture and keep aspect ratio

| | | |
|---|---|---|
| Syntax: | ReturnValue = ReadCWProperty(*TouchButtonVarName*, "**BackgroundPictureKeepAspectRatio**")<br>WriteCWProperty(*TouchButtonVarName*, "**BackgroundPictureKeepAspectRatio**", *Value*) | |
| Description: | Reads/sets, whether, when scaling (property "ScaleBackgroundPicture" = "TRUE") the aspect ratio of the background picture should be kept or not. | |
| Parameter: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return value | read value of the property (bool) |
| | Value | value to be set (bool): TRUE (default) or FALSE |

**Note**

See Chapter Positioning and aligning picture and text (Page 269).

## 7.6.4 Functions

You can call the subsequently listed functions using the CallCWMethod() function.

### Example

Setting the margins of the SIEsTouchButton linked using display variable "MyTouchButton".

The result is written to register 0.

```
REG[0] = CallCWMethod("MyTouchButton", "setMargins", 20, 20, 20, 20,
20)
```

### Overview

| Function | Description |
|---|---|
| setMargins | Setting the margins |
| serialize | Saves/restores the actual state |

### setMargins – setting the margins

| | |
|---|---|
| Syntax: | ReturnValue = CallCWMethod(*TouchButtonVarName*, "**setMargins**", *left*, *top*, *right*, *bottom*, *center*)<br>ReturnValue = CallCWMethod(*TouchButtonVarName*, "**setMargins**", *left*, *top*, *right*, *bottom*, *center*, *marginAlignment*) |
| Description: | Using this function, the margin clearances and the distance between picture and text are defined.<br>If a value of "-1" is specified, then this value is applicable for the system default clearance. If the value is greater or equal to "0", then the value is set as edge clearance. |

| | | |
|---|---|---|
| Parameters: | TouchButtonVarName | Name of the display variable that contains a SIEs-TouchButton |
| | Return Value | Error code (bool): TRUE = successful |
| | left | lefthand edge (int) |
| | top | top edge (int) |
| | right | right-hand edge (int) |
| | bottom | bottom edge (int) |
| | center | Clearance between picture and text (int), if the alignment is not "center" |
| | marginAlignment | Optional: Defines whether the margin next to the displayed picture should also apply when positioning the display text (bool), TRUE (default) |

### serialize – save/restore the actual state

| | | |
|---|---|---|
| Syntax: | ReturnValue = CallCWMethod(*TouchButtonVarName*, "**serialize**", *FilePath*, *IsStoring*) | |
| Description: | With this function, when required, you can write and restore the actual state and content of the SIEsTouchButton binary into a file or Data Stream. This function automatically refreshes the display. | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SIEsTouch-Button |
| | Return Value | Error code (bool): TRUE = successful |
| | FilePath | Complete path mit file name (QString) |
| | IsStoring | Save/restore (bool): TRUE = save |

**Note**

Programming engineers do not directly use this function!

## 7.6.5 Signals

You can capture the subsequently described signals in the configuration and respond appropriately.

### Overview

| Function | Description |
|---|---|
| clickedDisabled | Press a deactivated TouchButton |
| clicked | A click or tap is executed |
| checked | It was toggled |

- If a TouchButton can be operated - but it cannot be toggled - then "clicked" signal is sent when the TouchButton is pressed.
- If a TouchButton can be operated and toggled - then the following sequence of signals is sent when the TouchButton is pressed.
  "checked" → "clicked"
- If a TouchButton cannot be operated - then the "clickedDisabled" signal is sent when the TouchButton is pressed.

All of the signals mentioned above are only sent after an operator action, i.e. only once the mouse or the space bar has been released or after a tap for multitouch.

As a consequence, the SlEsTouchButton has a pure switch function, i.e. a pushbutton function is not possible.

**Note**

For multitouch operation, note that a click can only be sent if a tap gesture has been completely identified (pressing and releasing within approximately 0.7 seconds). If an action was already carried out when simply pressing, it would not be possible for example to scroll/shift the ScrollArea behind, or it could lead to operator input errors.

## clicked – the TouchButton was clicked

| | | |
|---|---|---|
| Syntax: | SUB(on_<TouchButtonVarName>_clicked)<br>…<br>END_SUB | |
| Description: | A complete sequence comprising pressing and releasing a TouchButton (that can be operated) results in a "clicked" signal. We recommend working predominantly with this signal. | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SlEsTouch-Button |
| | SIGARG[0] | Supplies the toggle state of the TouchButton (bool) |

## Example

```
DEF MyTouchButton = (W///,"slesstdcw.SlEsTouchButton"/////70,20,200,100/0,0,0,0)
SUB(on_MyTouchButton_clicked)
   DLGL("checked: " << SIGARG[0])
END_SUB
```

## checked – a TouchButton was toggled

| | | |
|---|---|---|
| Syntax: | SUB(on_<TouchButtonVarName>_checked)<br>…<br>END_SUB | |
| Description: | A TouchButton that can be toggled was pressed, which changed its state. | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SlEsTouch-Button |
| | SIGARG[0] | Supplies the toggle state of the TouchButton (bool) |

## Example

```
DEF MyTouchButton = (W///,"slesstdcw.SlEsTouchButton"/////70,20,200,100/0,0,0,0)
SUB(on_MyTouchButton_toggled)
   DLGL("toggled: " << SIGARG[0])
END_SUB
```

## clickedDisabled – a TouchButton that cannot be operated was clicked

| | | |
|---|---|---|
| Syntax: | SUB(on_<TouchButtonVarName>_clickedDisabled)<br>…<br>END_SUB | |
| Description: | A complete sequence comprising pressing and releasing a TouchButton (that cannot be operated) results in a "clickedDisabled" signal. | |
| Parameters: | TouchButtonVar-Name | Name of the display variable that contains a SlEsTouch-Button |
| | SIGARG[0] | Supplies the toggle state of the TouchButton (bool) |

## Example

```
DEF MyTouchButton = (W///,"slesstdcw.SlEsTouchButton"/////70,20,200,100/0,0,0,0)
SUB(on_MyTouchButton_clickedDisabled)
   DLGL("checkedDisabled: " << SIGARG[0])
END_SUB
```

## 7.6.6 Positioning and aligning picture and text

### Positioning pictures

A picture is positioned as follows with the specified alignment:

- In a first step, the picture matching the current resolution is determined from the associated resolution directory.
- The rectangle of the area (ClientArea) is then reduced by the specified clearances to the edges (MarginArea).
  The MarginArea can be influenced using the "setMargins" function:
  - setMargins(-1, -1, -1, -1, -1)

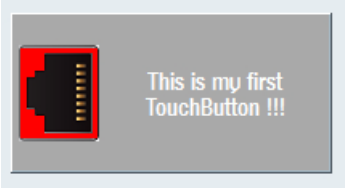



Figure 7-33 setMargins(-1, -1, -1, -1, -1)

  - setMargins(0, 0, 0, 0, 0)

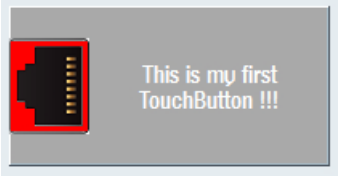



Figure 7-34 setMargins(0, 0, 0, 0, 0)

  - setMargins(20, 20, 20, 20, 20)

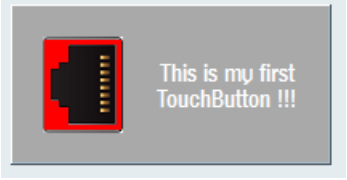



Figure 7-35 setMargins(20, 20, 20, 20, 20)

#### Example for alignment "left"

The area is horizontally halved, so that the picture to be displayed can assume half (maximum) of the area.

The picture is then drawn as follows:

- Property "scalePicture": FALSE
  The picture is drawn at this location without any scaling. It must be ensured that the basic picture is not too large, and can be completely displayed within the TouchButton.

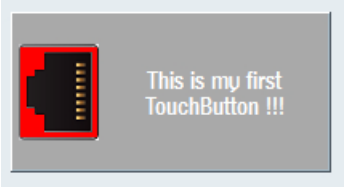



Figure 7-36 scalePicture FALSE

- Property "scalePicture": TRUE
  The picture is scaled (stretched or compressed), so that it fits into the left-hand half of the TouchButton. The "pictureKeepAspectRatio" property is taken into account as follows:
  - Property "pictureKeepAspectRatio": FALSE
    The picture is scaled horizontally and vertically so that it precisely fits into the left-hand half of the TouchButton. In so doing, the aspect ratio of the original picture is no longer taken into consideration.

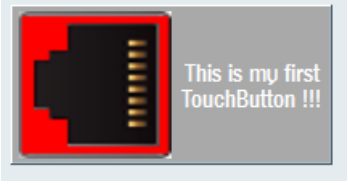



Figure 7-37 scalePicture - pictureKeepAspectRatio FALSE

  - Property "pictureKeepAspectRatio": TRUE
    Taking into consideration the aspect ratio of the original picture, the picture is scaled large enough so that it just fits into the left-hand half of the TouchButton.

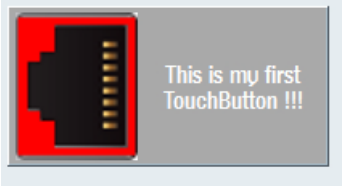



Figure 7-38 scalePicture - pictureKeepAspectRatio TRUE

**Note**

The same principle applies for "right", "top" and "bottom" alignments.

For the "centered" alignment, an attempt is made to fit the picture - depending on the "scalePicture" and "pictureKeepAspectRatio" properties - in the complete MarginArea.

### Positioning the text

The text is positioned as follows depending on the "textAlignedToPicture" property:

- Property "textAlignedToPicture": FALSE
  The text is shown in the MarginArea centered vertically and horizontally.

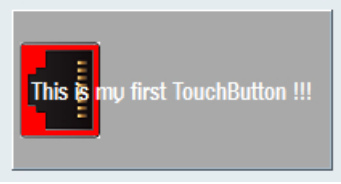



Figure 7-39 textAlignedToPicture FALSE

- Property "textAlignedToPicture": TRUE
  The text is output in the remaining area of the MarginArea minus the area of the drawn picture, centered horizontally and vertically.

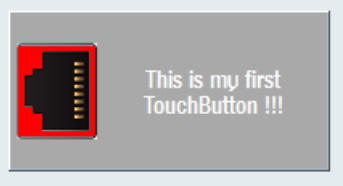



Figure 7-40 textAlignedToPicture TRUE

## 7.6.7 Language-dependent texts

SlEsTouchButton does not support foreign languages by itself. However, the following example shows how you can implement language dependency:

### Example

**easyscreen.ini:**

```
[LANGUAGEFILES]LngFile03 = user.txt
```

**user_eng.txt:**

```
85000 0 0 "This is my first TouchButton !!!"
```

**user_deu.txt:**

```
85000 0 0 "Das ist mein erster TouchButton !!!"
```

**Configuration file:**

```
//M(MyTBMask/"My CustomWidget TouchButton ...")
DEF MyTB1 = (W///,"slesstdcw.SlEsTouchButton"/////70,20,200,100/0,0,0,0)
LOAD
```

```
//M(MyTBMask/"My CustomWidget TouchButton ...")
DEF MyTB1 = (W///,"slesstdcw.SlEsTouchButton"/////70,20,200,100/0,0,0,0)
   WRITECWPROPERTY("MyTB1", "text", $85000)
END_LOAD
LANGUAGE
   WRITECWPROPERTY("MyTB1", "text", $85000)
END_LANGUAGE
```

The texts are automatically wrapped around at the word limits in the rectangle provided. The forced line break is only possible if the text to be displayed is sourced from a language file A "%n" must be inserted at the appropriate location in the text (see the following example).

**user_eng.txt:**

```
85001 0 0 "This is%nmy%nfirst%nTouchButton !!!"
```

**Configuration file:**

```
WRITECWPROPERTY("MyTB1", "text", $85001)
```

## Result




Figure 7-41 Example of a line break in a language-dependent text

# "Custom" operating area

# 8

## 8.1 How to activate the "Custom" operating area

### Activating the "Custom" operating area

The "Custom" operating area is not activated on delivery.

1. First copy the "slamconfig.ini" file from the *[System Siemens directory]*\cfg directory to the *[System oem directory]*\cfg directory
   or correspondingly to the *[System add_on directory]*\cfg directory or *[System user directory]*\cfg directory.
2. To activate the "Custom" operating area, the following must be entered:
   ```
   [Custom]
   Visible=True
   ```

### Result

After activation is complete, the softkey for the "Custom" operating area can be found in the main menu (F10) on the menu continuation bar on the HSK4 (= default).

The "Custom" operating area displays an empty window covering the entire operating area, with a configurable header. All horizontal and vertical softkeys can be configured.

## 8.2 How to configure the "Custom" softkey

### Configuring the softkey for the "Custom" operating area

The labeling and position of the softkey for the "Custom" operating area are configured in the "slamconfig.ini" file.

The following options are available for configuring the start softkey:

1. To replace a softkey label with a **language-dependent text**, the following must be entered in the [Custom] section:
   ```
   TextId=MY_TEXT_ID
   TextFile=mytextfile
   TextContext=mycontext
   ```
   In this example, the softkey shows the language-dependent text which was saved with the text ID "MY_TEXT_ID" in text file mytextfile_xxx.qm under "MyContext" (xxx stands for language code).
2. To replace a softkey label with a **language-neutral text**, the following must be entered in the [Custom] section:
   ```
   TextId=HELLO
   TextFile=<empty>
   TextContext=<empty>
   ```
   In this example, the softkey for the "Custom" operating area displays the text "HELLO" for every language.
3. **An icon** can also be displayed on the softkey in addition to the text.
   To do this, the following must be entered in the [Custom] section:
   ```
   Picture=mypicture.png
   ```
   The softkey then displays the icon from the file mypicture.png. Graphics and bitmaps are stored at the following path: *[System oem directory]*ico/ico<resolution>. The directory that corresponds to the display resolution must be used.
4. **The position** of the softkey can also be set. The following entry in the [Custom] section makes this setting:
   ```
   SoftkeyPosition=12
   ```
   The default is position 12. This corresponds to the HSK4 on the menu continuation bar of the operating area's menu. Positions 1 - 8 correspond to HSK1 to HSK8 on the menu bar, positions 9 - 16 to HSK1 to HSK8 on the menu continuation bar.

# 8.3 How to configure the "Custom" operating area

## Configuring the softkey for the "Custom" operating area

You need the "easyscreen.ini" and "custom.ini" files to configure the operating area. Templates for both these files are available in the *[System Siemens directory]*/templates/cfg directory.

1. First copy the files to the *[System oem directory]*/cfg directory and make your changes from there.
2. The "easyscreen.ini" file already contains a definition line for the "Custom" operating area:
   ```
   ;StartFile02 = area := Custom, dialog := SlEsCustomDialog,
   startfile := custom.com
   ```
   The ";" at the start of the line represents the comment character. This means the line is commented out and, as such, not active. To change this, the ";" must be deleted.
   The "startfile" attribute in this line defines that the entry will refer to the "custom.com" project file when the "Custom" operating area is selected.
3. You create the **project file "custom.com"** in the *[System oem directory]*/proj directory. This contains the relevant configuration which is created in the same way as the "aeditor.com" file of the "Program" operating area. The configured start softkeys are then displayed in the "Custom" operating area.
4. You configure the **language-neutral text** for the title bar of the dialog in the "custom.ini" file. The following entry is available in the template for this purpose:
   ```
   [Header]
   Text=Custom
   ```
   You can replace this text with a customized one.
5. The template contains the following entry for configuring a **start screen** for the "Custom" operating area:
   ```
   [Picture]
   Picture=logo.png
   ```
   Logo.png is the name of the start screen which appears on the "Custom" operating area's start dialog. Here you can display a company logo, for example, or another image. The file should be saved in the directory for the corresponding resolution under: *[System oem directory]*/ico/ …
6. In order to display a specific "Run MyScreens" screen when the "Custom" operating area is shown the first time, the following entry is available in the template:
   ```
   ; Mask shown with startup of area "custom"
   [Startup]
   ;Startup = Mask:=MyCustomStartupMask, File:=mycustommasks.com
   ```
   Typically, SINUMERIK Operate is started directly with a specific "Run MyScreens" screen. The "startuparea" key must be set as follows in the "[miscellaneous]" section of the "systemconfiguration.ini" file:
   ```
   [miscellaneous]
   ;name of the area to be shown at system startup
   startuparea = Custom
   ```

## 8.4 Programming example for the "Custom" area

### Example

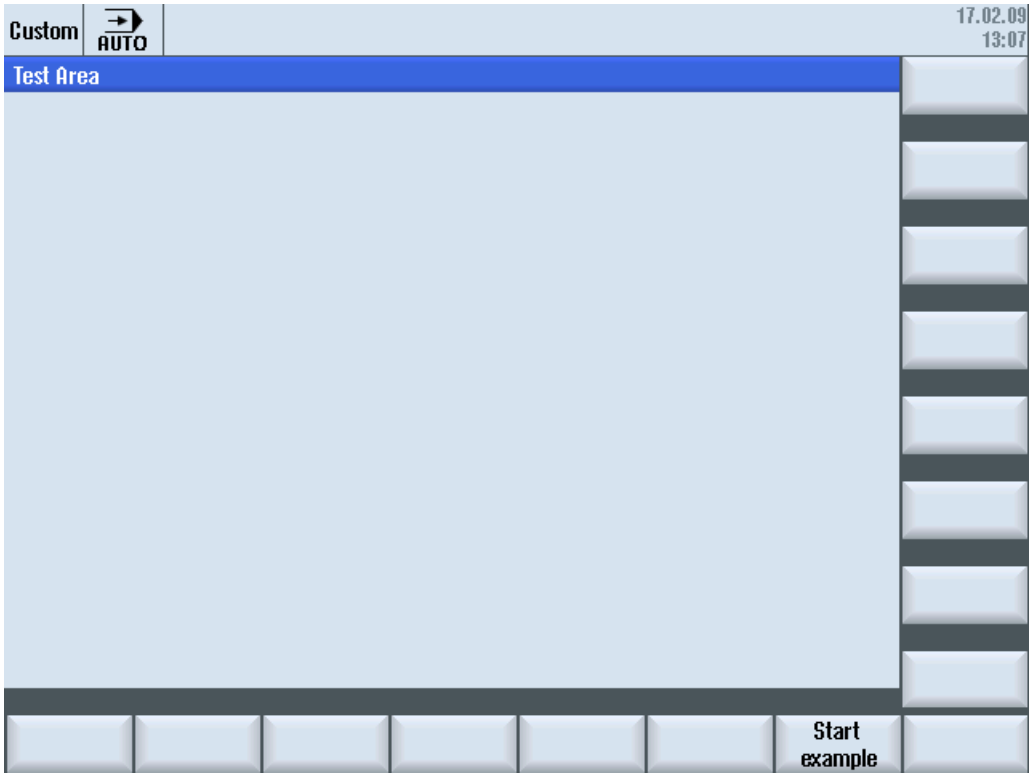



Figure 8-1 Example with "Start example" softkey

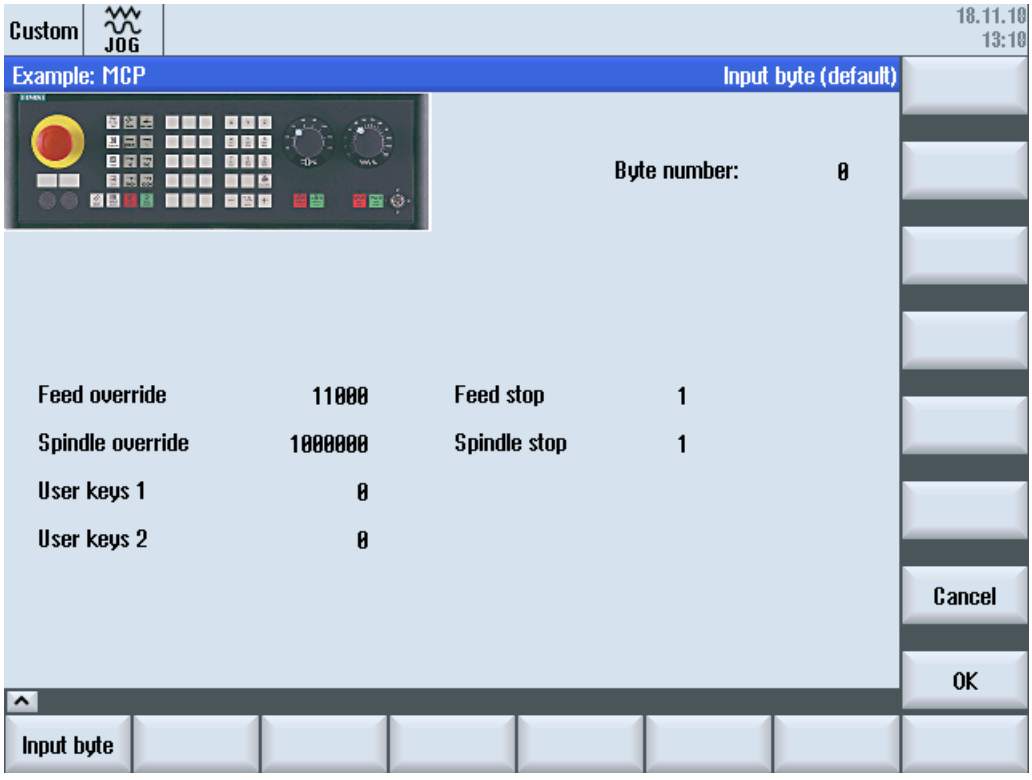



Figure 8-2 Example with bitmap and text fields

## File overview

The following files are required:

- "custom.com"
- "easyscreen.ini"

## Programming

Content of file "custom.com":

**Note**

The graphic file "mcp.png" integrated in the example is also only a sample file. If you wish to use this programming example in your application, you must replace the graphic by one of your own graphics.

```
//S(Start)
   HS7=("Start example", se1, ac7)
   PRESS(HS7)
      LM("Maske4")
```

```
   END_PRESS
//END
//M(Maske4/"Example: MCP"/"mcp.png")
   DEF byte=(I/0/0/"Input byte=0 (default)","Byte number:",""/wr1,li1///380,40,100/480,40,50)
   DEF Feed=(IBB//0/"","Feed override",""/wr1//"EB3"/20,180,100/130,180,100), Axistop=(B//0/"","Feed
stop",""/wr1//"E2.2"/280,180,100/380,180,50/100)
   DEF Spin=(IBB//0/"","Spindle override",""/wr1//"EB0"/20,210,100/130,210,100), spinstop=(B//
0/"","Spindle stop",""/wr1//"E2.4"/280,210,100/380,210,50/100)
   DEF custom1=(IBB//0/"," User keys 1",""/wr1//"EB7.7"/20,240,100/130,240,100)
   DEF custom2=(IBB//0/"","User keys 2",""/wr1//"EB7.5"/20,270,100/130,270,100)
   DEF By1
   DEF By2
   DEF By3
   DEF By6
   DEF By7

   HS1=("Input byte", SE1, AC4)
   HS2=("")
   HS3=("")
   HS4=("")
   HS5=("")
   HS6=("")
   HS7=("")
   HS8=("")
   VS1=("")
   VS2=("")
   VS3=("")
   VS4=("")
   VS5=("")
   VS6=("")
   VS7=("Cancel", SE1, AC7)
   VS8=("OK", SE1, AC7)

   PRESS(VS7)
      EXIT
   END_PRESS

   PRESS(VS8)
      EXIT
   END_PRESS

   LOAD
      By1=1
      By2=2
      By3=3
```

```
      By6=6
      By7=7
   END_LOAD

   PRESS(HS1)
      Byte.wr=2
   END_PRESS

   CHANGE(Byte)
      By1=byte+1
      By2=byte+2
      By3=byte+3
      By6=byte+6
      By7=byte+7
      Feed.VAR="EB"<<By3
      Spin.VAR="EB"<<Byte
      Custom1.VAR="EB"<<By6
      Custom2.VAR="EB"<<By7
      Axisstop.VAR="E"<<By2<<".2"
      Spinstop.VAR="E"<<By2<<".4"
      Byte.wr=1
   END_CHANGE

   CHANGE(Axis stop)
      IF Axistop==0
         Axistop.BC=9
      ELSE
         Axistop.BC=11
      ENDIF
   END_CHANGE

   CHANGE(Spin stop)
      IF Spinstop==0
         Spinstop.BC=9
      ELSE
         Spinstop.BC=11
      ENDIF
   END_CHANGE
//END
```

# Dialog selection

# 9

## 9.1 Dialog selection using PLC softkeys

### Configuration

Description of the procedure:

- The "systemconfiguration.ini" contains a section [keyconfiguration]. The entry specifies an action for a special PLC softkey.
- A number is given as an action. A "Run MyScreens" call is involved if the number is greater than or equal to 100.
- A section for defining the action to be performed must be created in the "easyscreen.ini" file. The name of the section is based on the name of the operating area and the dialog name (see entry under [keyconfiguration] → Area:=..., Dialog:=...) → [<Area>_<Dialog>] → e.g. [AreaParameter_SlPaDialog]
- The action numbers (which were given in the "systemconfiguration.ini" → see Action:=...) are defined in this section. There are two commands involved:
  1. LS("Softkey menu1","param.com") ... Loading a softkey menu
  2. LM("Screen form1","param.com") ... Loading a screen form

### Selecting softkey menus via PLC softkeys

"Run MyScreens" makes it possible to select "Run MyScreens" softkey menus and "Run MyScreens" dialogs via PLC softkeys. This can only be done if the "action" attribute to be specified when configuring the relevant PLC softkeys has a value greater than or equal to 100.

PLC softkeys are configured in the file "systemconfiguration.ini" in the section [keyconfiguration]:

```
[keyconfiguration]
KEY75.1 = Area:=area, Dialog:=dialog, Screen:=screen, Action:= 100,
```

The LM and LS commands to be executed upon activation of the relevant PLC softkeys are configured in the "easyscreen.ini" file. The names of the sections that are used for the purpose of configuration are structured as follows:

| | |
|---|---|
| `[areaname_dialogname]` | The first part of the name "areaname" refers to the operating area and the second part "dialogname" designates the dialog to which the commands configured in this section apply. |
| `[AreaParameter_SlPaDialog]`<br>`100.screen1 = LS("Softkey1","param.com")`<br>`101.screen3 = LM("Screen form1","param.com")` | The names given in the "systemconfiguration.ini" file for the operating area and dialog should be used. The dialog does not have to be specified.<br>This is particularly true for operating areas which are only implemented by means of a single dialog. Please refer to the example on the left.<br>If "screen1" is displayed in the AreaParameter operating area implemented by the SlPaDialog dialog, the "LS("Softkey1","param.com")" command will be executed when the "action" with the value 100 occurs. |
| `action.screen=Command` | Both the "action" and "screen" attributes clearly indicate when the specified command will be executed.<br>The "screen" information is optional.<br>The following commands are permissible:<br>`LM (LoadMask)`<br>`LS (LoadSoftkeys)` |

# 9.2 Dialog selection using PLC hard keys

## Field of application

The following functions can be initiated in the operating software by the PLC:

- Select an operating area
- Select certain scenarios within operating areas
- Execute functions configured at softkeys

## Hardkeys

All keys - also the PLC keys - are subsequently referred to as hardkeys. A maximum of 254 hardkeys can be defined. The following allocation applies:

| Key number | Application |
|---|---|
| Key 1 – key 9 | Keys on the operator panel front |
| Key 10 – key 49 | Reserved |
| Key 50 – key 254<br>Key 50 – key 81 | PLC keys:<br>Reserved for OEMs |
| Key 255 | Pre-assigned by control information. |

Hardkeys 1 - 9 are pre-assigned as follows:

| Key designation | | Action / effect |
|---|---|---|
| HK1 | MACHINE | Selects "Machine" operating area, last dialog |
| HK2 | PROGRAM | Selects "Program" operating area, last dialog or last program |
| HK3 | OFFSET | Selects "Parameter" operating area, last dialog |
| HK4 | PROGRAM MANAG-ER | Selects "Program" operating area, "Program Manager" basic screen |
| HK5 | ALARM | Selects "Diagnostics" operating area, "Alarm list" dialog |
| HK6 | CUSTOM | Selects "Custom" operating area |
| HK7 1) | MENU SELECT | Selects "Main menu" |
| HK8 1) | MENU FUNCTION | Selects "Function" operating area |
| HK9 1) | MENU USER | Selects "User" operating area |

1) For 828D only

## Configuration

The configuring is realized in the "systemconfiguration.ini" configuration file in the section [keyconfiguration]. Each line defines what is known as a hardkey event. A hardkey event is the n-th actuation of a specific hardkey. For example, the second and third actuation of a specific hardkey can result in different responses.

The entries in the "systemconfiguration.ini" configuration file can be overwritten with user-specific settings. The directories *[System user directory]*/cfg and [System oem directory]/cfg are available for this purpose.

The lines for configuring the hardkey events have the following structure:

```
KEYx.n = area:=area, dialog:=dialog, screen:=screen, forms:=form,
menus:=menu,
action:=menu.action, cmdline:=cmdline
KEYx.n = area:=area, dialog:=dialog, cmdline:=cmdline, action:=
action
```

x: Number of the hardkey, range of values: 1 – 254

n: Event number – corresponds to the nth actuation of the hardkey, range of values: 0 – 9

## Requirement

The PLC user program must fulfill the following requirement:

Only one hardkey is processed. As a consequence, a new request can only be set if the operating software has acknowledged the previous request. If the PLC user program derives the hardkey from an MCP key, it must provide sufficient buffer storage of the key(s) to ensure that no fast keystrokes are lost.

## PLC interface

An area to select a hardkey is provided in the PLC interface. The area is in DB19.DBB10. Here, the PLC can directly specify a key value of between 50 and 254.

Acknowledgment by the operating software takes place in two steps. This procedure is necessary so that the operating software can correctly identify two separate events if the same key code is entered twice consecutively. In the first step, control information 255 is written to byte DB19.DBB10. This defined virtual key stroke enables the HMI to identify every PLC key sequence uniquely. The control information is of no significance to the PLC user program and must not be changed. In the second step, the actual acknowledgment takes place with respect to the PLC by clearing DB19.DBB10. From this point in time, the PLC user program can specify a new hardkey. In parallel, the actual hardkey request is processed in the operating software.

## Example

Configuration file:

```
; PLC hardkeys (KEY50-KEY254)
[keyconfiguration]
KEY50.0 = name := AreaMachine, dialog := SlMachine
KEY51.0 = name := AreaParameter, dialog := SlParameter
KEY52.0 = name := AreaProgramEdit, dialog := SlProgramEdit
KEY53.0 = name := AreaProgramManager, dialog := SlPmDialog
KEY54.0 = name := AreaDiagnosis, dialog := SlDgDialog
KEY55.0 = name := AreaStartup, dialog := SlSuDialog
KEY56.0 = name := Custom, dialog := SlEsCustomDialog
```

The area and dialog identifiers can be found in the "systemconfiguration.ini" from *[System Siemens directory]*\cfg.

# 9.3 Dialog selection via NC

## MMC command in HMI Operate

You can use MMC commands as described in the following:

1. **Definition of MMC commands**
   The following combinations are available in the standard "systemconfiguration.ini" file:
   address:=MCYCLES --> command:=LM
   address:=CYCLES --> command:=PICTURE_ON
   This distinction is necessary, in order to make a differentiation between measuring cycles and other cycles. This means that:
   - LM is always applicable for measuring cycles, PICTURE_ON always for non-measuring cycles
   - It is not permissible that newly defined MMC commands are designated PICTURE_ON and LM
2. **"Run MyScreens" license**
   All of the dialogs opened by "Run MyScreen" – with the exception of the measuring cycles – fall under the "Run MyScreens" license, and therefore without license, only a limited number of dialogs can be used.
   Example call with "test.com" (Run MyScreens):

```
g0 f50
MMC("CYCLES,PICTURE_ON,test.com,screen form1","A")
m0
MMC("CYCLES,PICTURE_OFF","N")
M30
```

# Examples for cycle masks 10

## 10.1 Examples for cycle masks

When installing SINUMERIK Operate, examples for cycle masks created with Run MyScreens are also saved to the system.

The Run MyScreens examples are available under the following archive paths:

| | | |
|---|---|---|
| *[System Siemens directory]* | | |
| ...\siemens\sinumerik\hmi\template\easyscreen\ | | |
| ... | | |
| | standard\ | Help screens, PNG |
| | x3d\ | Help screens / animations |
| ... | | |
| | Milling\ | Fräsbearbeitung |
| | Turning\ | Drehbearbeitung |
| ... | | |
| | deu\ | German description |
| | eng\ | English description |

Examples in the commissioning operating area can be found on the HMI user interface:

Horizontal softkey system data → HMI data → Templates → Examples → Easy Screen → … (see above).

### Brief description of the examples

- Milling
  - Drilling example: This is where a position can be added
  - Measure workpiece example: This is where a cycle call can be generated and processed via NC start while the dialog is open
  - Workpiece counter example: This shows how GUDs should be handled
- Turning
  - Fixtures: this contains a tailstock, bar loader, part catcher
  - Measure workpiece example: This is where a cycle call can be generated and processed via NC start while the dialog is open
  - Drilling example: This is where a position can be added

# Configuration in the side screen

# 11

The following options are available to you for displaying Run MyScreens configurations in the side screen:

1. Display of a Run MyScreens configuration in a (separate) side screen page:
   The Run MyScreens configuration takes up all of the space of the side screen page.
2. Display of several Run MyScreens configurations in a (separate) side screen page:
   The individual Run MyScreens configurations are displayed above each other within the side screen page, in so-called side screen elements. The side screen elements, and thus also the Run MyScreens configurations, can be scrolled vertically.
3. Adding one or more Run MyScreens configurations to an existing side screen page:
   I.e. the Run MyScreens configuration is added to a side screen page contained in the SIEMENS delivery kit. This case matches point 2 with the exception of the affected side screen page.
4. Adding one or more Run MyScreens configurations to a side screen element:
   The individual Run MyScreens configurations are displayed next to each other in so-called side screen widgets and can be scrolled horizontally.

# 11.1 Display of a Run MyScreens configuration in a side screen page

To display a Run MyScreens configuration in a (separate) side screen page, you must add a new side screen page to the existing side screen configuration ("slsidescreen.ini"). The Run MyScreens configuration is then displayed or executed in this side screen page.

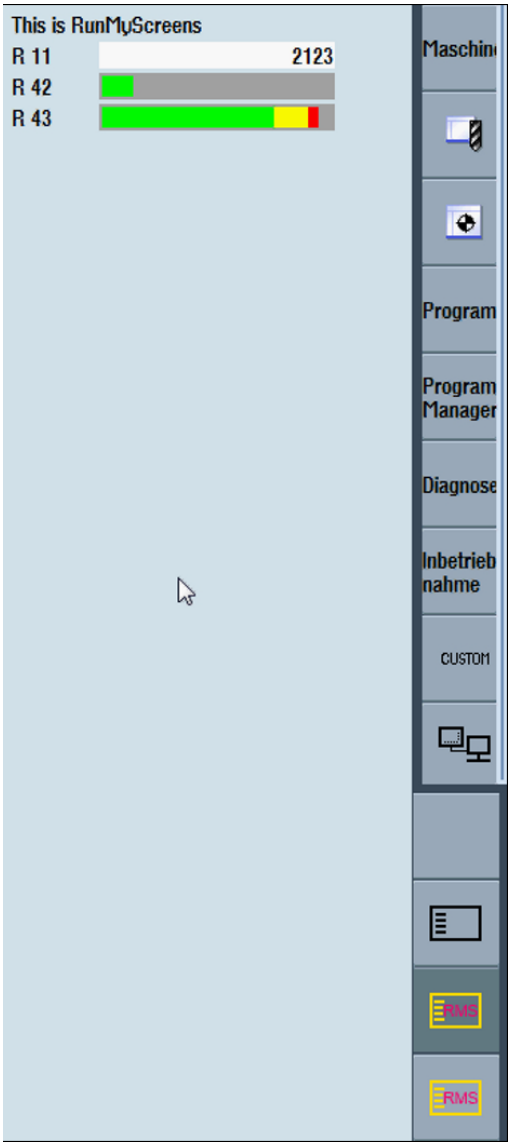



Figure 11-1 **Display of a Run MyScreens configuration in a side screen page**

## Procedure

Expand the file "slsidescreen.ini" by the following entry:

```
[Sidescreen]
PAGE100= name:=RMSPage,
implementation:=sleseasyscreen.SlEsSideScreenPage
```

- The number of the page, "100" in this case, must be counted up corresponding to existing side screen pages. The number of the page must be ≥ 100. This avoids conflicts with the SIEMENS pages.
- You can freely select the name of the page, "RMSPage" in this case.
- You cannot change the specification for implementing the page, "sleseasyscreen.SlEsSideScreenPage" in this case.

Configure the Run MyScreens configuration that is to be carried out in the next step. To do this, create a new section with a name that contains the name of the newly created page:

```
[Page_RMSPage]
Icon=rmspage.png
PROPERTY001= name:=maskPath, type:=QString,
value:="sidescreenmask.com"
PROPERTY002= name:=maskName, type:=QString, value:="Mask"
PROPERTY003= name:=focusable, type:=bool, value:="true"
```

- The name of the section is formed by appending the name of the page to the prefix "Page_", in this case "RMSPage".
- Under Icon, specify the file name, "rmspage.png" in this case. The file contains the icon for the button for selecting the page. Store the file in the directory */user/sinumerik/hmi/ico/ico640* or */oem/sinumerik/hmi/ico/ico640*.
- In the property maskPath, specify the name of the file that contains the RunMyScreens configuration, "sidescreenmask.com" in this case.
- In the property maskName, specify the name of the RunMyScreens mask that is to be displayed, "sidescreenmask.com" in this case.
- Specify the input focus in the Property focusable. The page can be assigned the input focus only if this attribute has the value "true". This attribute is required for pages with input elements.

Store the "slsidescreen.ini" in the directory */oem/sinumerik/hmi/cfg* or */user/sinumerik/hmi/cfg*.

The Run MyScreens configuration is available after the next HMI power-up.

## Example

Example for a complete configuration in "slsidescreen.ini":

```
[Sidescreen]
PAGE005= name:=RMSPage,
implementation:=sleseasyscreen.SlEsSideScreenPage
[Page_RMSPage]
PROPERTY001= name:=maskPath, type:=QString,
value:="sidescreenmask.com"
PROPERTY002= name:=maskName, type:=QString, value:="Mask"
```

## 11.2 Display of several Run MyScreens configurations in a side screen page

For the display of several Run MyScreens configurations in a (separate) side screen page, you must also configure so-called side screen elements, in addition to the side screen page. The side screen elements are used to display the Run MyScreens configurations.

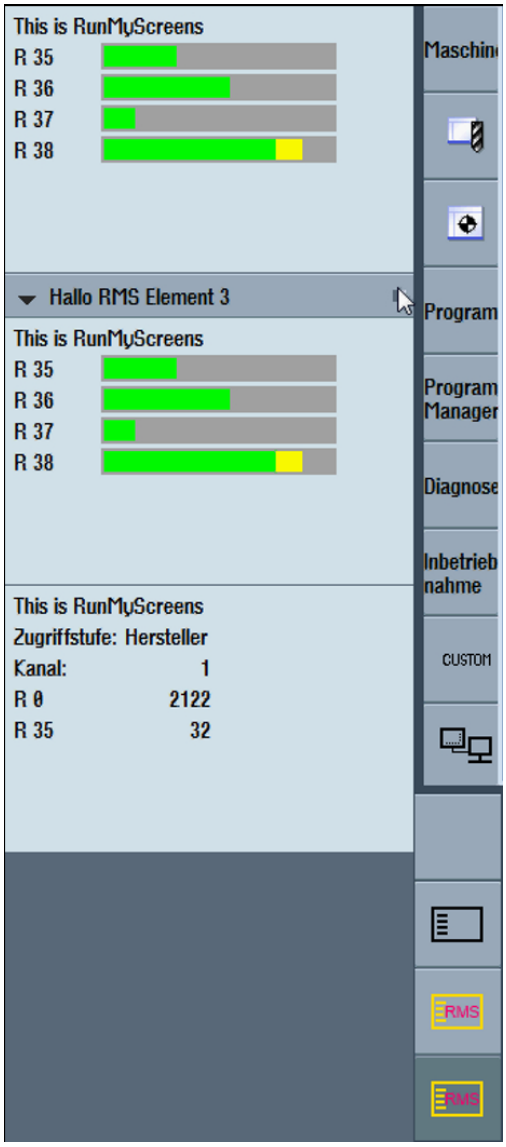



Figure 11-2 **Display of several Run MyScreens configurations in a side screen page**

### Procedure

Expand the "slsidescreen.ini" by the following entries:

```
[Sidescreen]
PAGE100= name:=RMSPage, implementation:=SlSideScreenPage
```

- The number of the page, "100" in this case, must be counted up corresponding to existing side screen pages. The number of the page must be ≥ 100. This avoids conflicts with the SIEMENS pages.
- You can freely select the name of the page, "RMSPage" in this case.
- You cannot change the specification for implementing the page, "SlSideScreenPage" in this case.

```
[Page_RMSPage]
ELEMENT001= name:=RMSElement001,
implementation:=sleseasyscreen.SlEsSideScreenElement
ELEMENT002= name:=RMSElement002,
implementation:=sleseasyscreen.SlEsSideScreenElement
```

- Two side screen elements are assigned to the page – RMSElement001 and RMSElement002. You can freely select the names of the elements.
- You cannot change the specification for implementing the elements, "sleseasyscreen.SlEsSideScreenElement" in this case.

Finally, you must assign Run MyScreens configurations that are still to be displayed in these elements to the side screen elements:

```
[Element_RMSElement001]
PROPERTY001= name:=maskPath, type:=QString,
value:="sidescreenmask.com"
PROPERTY002= name:=maskName, type:=QString, value:="Mask001"
…
[Element_RMSElement002]
PROPERTY001= name:=maskPath, type:=QString,
value:="sidescreenmask.com"
PROPERTY002= name:=maskName, type:=QString, value:="Mask002"
```

Store the "slsidescreen.ini" in the directory */oem/sinumerik/hmi/cfg* or */user/sinumerik/hmi/cfg*.

The Run MyScreens configuration is available after the next HMI power-up.

## 11.3 Adding Run MyScreens configurations to a side screen page

The procedure for integrating one or several Run MyScreens configurations for an existing side screen page is identical to the procedure described in Chapter Display of several Run MyScreens configurations in a side screen page (Page 292). The only difference is that the side screen elements for displaying the Run MyScreens configurations are added to an existing side screen page.

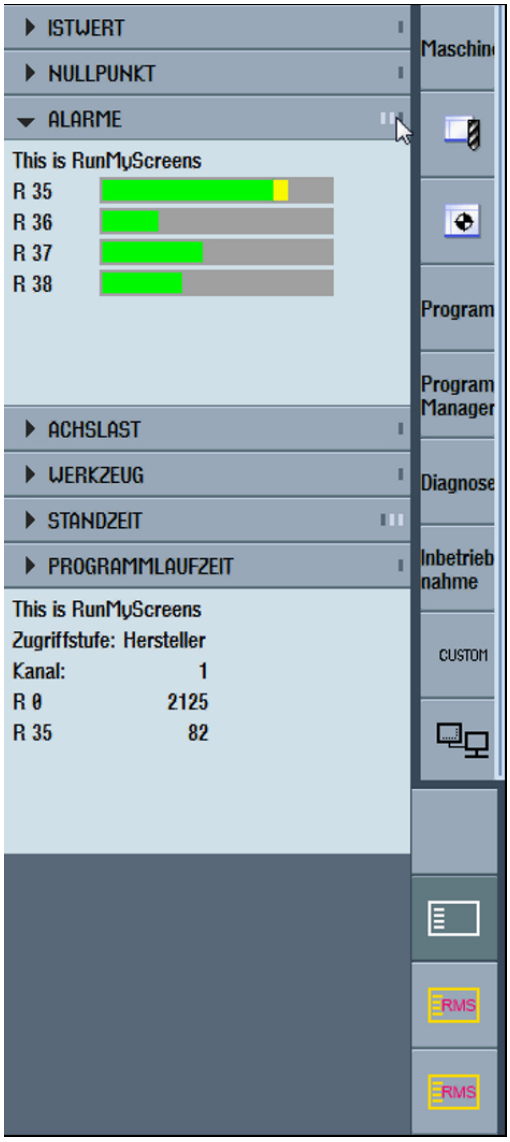



Figure 11-3 **Adding Run MyScreens configurations to a side screen page**

**Note**

Of the side screen pages contained in the SIEMENS delivery kit, you can only expand the WidgetsPage (see file "slsidescreen.ini") with Run MyScreens configurations.

## Procedure

Expand the "slsidescreen.ini" by the following entries:

In the [Page_WidgetsPage] section, add a corresponding element for receiving the Run MyScreens configuration.

```
[Page_WidgetsPage]
ELEMENT100= name:=RMSElement,
implementation:=sleseasyscreen.SlEsSideScreenElement
```

- Use a number range ≥ 100 for new elements. This avoids conflicts with the SIEMENS display elements. The RMSElement is inserted in the WidgetsPage under the SIEMENS elements according to this numbering.

```
[Element_RMSElement]
PROPERTY001= name:=maskPath, type:=QString,
value:="sidescreenmask.com"
PROPERTY002= name:=maskName, type:=QString, value:="Mask001"
```

## 11.4 Adding Run MyScreens configurations to a side screen element

The basis of the procedure that is described in the following as example is a side screen page with a side screen element. Two Run MyScreens configurations are displayed next to each other in the side screen element.

Configure a side screen page with a side screen element as described in the following. Assign two side screen widgets with the respective Run MyScreens configurations to the side screen element.

### Procedure

Expand the "slsidescreen.ini" by the following entries:

```
[Sidescreen]
PAGE100= name:=RMSPage, implementation:=SlSideScreenPage
```

- You cannot change the specification for the implementation, "SlSideScreenPage" in this case.

```
[Page_RMSPage]
ELEMENT001= name:=RMSElement, implementation:= SlSideScreenElement
```

- You cannot change the specification for the implementation, "SlSideScreenElement" in this case.

```
[Element_RMSElement]
TextId=RMS_ELEMENT_TITLE
TextFile=rmstexts
TextContext=rmstexts
TextContext=rmstexts
WIDGET001= name:=RMSWidget001,
implementation:=sleseasyscreen.SlEsSideScreenWidget
WIDGET002= name:=RMSWidget002,
implementation:=sleseasyscreen.SlEsSideScreenWidget
```

- When the SlSideScreenElement implementation is used for the side screen element (see Section [Page_RMSPage]), the side screen element has a title bar in which a text can be displayed.
  This text is configured with the entries TextId, TextFile and TextContext:
  - TextId: Text ID of the text to be displayed
  - TextFile: File in which the text is saved
  - TextContext: Text context, in which the text is located within this file

  A separate file must be provided for each language.
  The name of these files is formed according to the following diagram:
  <TextFile>_<Sprachsuffix>.ts, e.g. rmstexts_deu.ts for the example above.
  If you do not specify a file or a context (TextFile=<empty> and TextContext=<empty>), the text ID is displayed as text. You can specify any string as the text ID.
  The text file (e.g. rmstexts_deu.ts) is structured as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS>
<TS>
 <context>
 <name>rmstexts</name>
 <message>
 <source>My application</source>
 <translation>My Application</translation>
 </message>
 </context>
</TS>
```

  Store the file in the directory */oem/sinumerik/hmi/lng* or */user/sinumerik/hmi/lng*. During the next HMI power-up, the file is converted to the file format that is needed during runtime.
- You can freely select the names of the widgets assigned to the element, "RMSWidget001" and "RMSWidget002" in this case.
- You must not change the widget implementation "sleseasysscreen.SlEsSideScreenWidget".

```
[Widget_RMSWidget001]
PROPERTY001= name:=maskPath, type:=QString, value:="mymask001.com"
PROPERTY002= name:=maskName, type:=QString, value:="MyMask001"

[Widget_RMSWidget002]
PROPERTY001= name:=maskPath, type:=QString, value:="mymask002.com"
PROPERTY002= name:=maskName, type:=QString, value:="MyMask002"
```

- The names of the sections are formed for configuring the side screen widgets by appending the name of the widget that is to be configured in this section to the prefix "Widget_".
- In the Property maskPath, specify the name of the file that contains the RunMyScreens configuration, "mymask001.com" or "mymask002.com" in this case.
- In the Property maskName, specify the name of the RunMyScreens mask that is to be displayed, "MyMask001" or "MyMask002" in this case.

## 11.5 Notes on carrying out Run MyScreens configurations in the side screen

Note the following:

- The functions LS, LM, DLGL, EXIT, EXITLS are not available when carrying out Run MyScreens configurations in the side screen.
- It is not possible to query the mask size in the LOAD block.
- Texts are always displayed using the font that is used in SINUMERIK Operate with a screen resolution of 640x480 pixels.
- The configured positions of operator controls are implemented without changes (e.g. extension).
- The dialog size is adapted automatically.
- If Run MyScreens configurations are carried out in a side screen page, the entire area of the page is available for the configuration. When carrying out in side screen elements or widgets, the area available for display is specified by the system depending on the context.
- The header and softkeys cannot be configured.
- All of the debugging and error outputs are sequentially written to the text file "easyscreen_log.txt". There is no specific identification indicating which configuration a message comes from.
- The status of a Run MyScreens configuration displayed in the side screen is rejected when the configuration is faded out.

**Note**

The integration of Run MyScreens configurations into the side screen can lead to more than one Run MyScreens configuration being carried out simultaneously. To maintain the functional capability of the overall system, take note of the additional system load caused as a result of the hardware used.

# Configuration in the Display Manager 12

## 12.1 Displaying Run MyScreens configurations in the Display Manager

The SISideScreenDialog dialog is available to you for displaying Run MyScreens configurations in the Display Manager. Depending on the space available, this dialog can display one or more side screen pages (see IM9 Chapter 3.13). Several pages are displayed next to each other in column format. The available space (width) is evenly divided among the individual columns. The number of pages to be displayed and their content are configured. Each page has its own configuration file. The names of these configuration files are specified during the dialog configuration in the "systemconfiguration.ini" file, in the command line parameter "cmdline". Behind the parameter identifier "–sidescreen1" is the name of the configuration file for the first column; behind the parameter identifier "–sidescreen2" is the name of the configuration file for the second column, etc.

### Example

```
[dialogs]
DLG500= name:=RMSApp,
implementation:=sldmsidescreenapp.SlSideScreenDialog,
process:=SlHmiHost1, preload:=false, cmdline:="-sidescreen1
rmspage1.ini –sidescreen2 rmspage2.ini –spacing 3″
```

The instance of the SlSideScreenDialog dialog created in this example has the name RMSApp. It has 2 pages/columns. The two pages/columns are configured in the files "rmspage1.ini" and "rmspage2.ini". Between the two columns, there is a distance of 3 pixels.

**Note**

You can freely select the names of the files with the page configurations, "rmspage1.ini" and "rmspage2.ini" in this case.

The use of Run MyScreens configurations in the Display Manager is done as described in Chapter Configuration in the side screen (Page 289). The only difference is that the integration of the configurations is not done in the file "slsidescreen.ini", but in the respective files provided for the configuration of the existing pages.

2 variants are available to you for the integration of Run MyScreens configurations into the Display Manager of SINUMERIK Operate. These are described in the following sections.

## 12.2 Integration in SISideScreenDialog

For the integration of the Run MyScreens configuration, you must create the dialog instance of SlSideScreenDialog in the file "systemconfiguration.ini" and create the file for integrating the Run MyScreens configuration.

The following example shows the integration of a Run MyScreens configuration. The Run MyScreens configuration takes up the entire window of the SlSideScreenDialog dialog, i.e. only one page/column exists.

### Procedure

First, create an instance of the SlSideScreenDialog in the "systemconfiguration.ini" file in the `[dialogs]` section. The instance is given the name "RMSApp".

```
[dialogs]
DLG500= name:=RMSApp,
implementation:=sldmsidescreenapp.SlSideScreenDialog,
process:=SlHmiHost1, preload:=false,
cmdline:="-sidescreen1 rmsapp.ini"
```

In the next step, you will configure or integrate the RunMyScreens configuration in the "rmsapp.ini" file:

```
[Sidescreen]
PAGE001= name:=RMSPage, implementation:=SlSideScreenPage
```

- Create a page with the name "RMSPage".

```
[Page_RMSPage]
ELEMENT001= name:=RMSElement,
implementation:=sleseasyscreen.SlEsSideScreenElement
```

- You create an element with the name RMSElement on the RMSPage. This element is used for displaying the Run MyScreens configuration.

```
[Element_RMSElement]
PROPERTY001= name:=maskPath, type:=QString,
value:="sidescreenmask.com"
PROPERTY002= name:=maskName, type:=QString, value:="Mask"
```

- In the property maskPath, specify the name of the file that contains the RunMyScreens configuration, "sidescreenmask.com" in this case.
- In the property maskName, specify the name of the RunMyScreens mask that is to be displayed, "sidescreenmask.com" in this case.

Store the files "rmsapp.ini" and "systemconfiguration.ini" (see above) in the directory */oem/sinumerik/hmi/cfg* or */user/sinumerik/hmi/cfg*.

The Run MyScreens configuration is available after the next HMI power-up.

## 12.3 Integration into SIWidgetsApp

The integration of a Run MyScreens configuration into the SIWidgetsApp application, which is part of the SIEMENS delivery kit, is described in the following.

Depending on which page/column of the SIWidgetsApp application the Run MyScreens configuration is to be integrated into, you must expand either the "sldmwidgets1.ini" file or the "sldmwidgets2.ini" file accordingly.

The following example shows the integration of a Run MyScreens configuration into the left page/column of the SIWidgetsApp application.

### Procedure

Expand the "sldmwidgets1.ini" file as follows:

```
[Sidescreen]
PAGE001= name:=WidgetsPage, implementation:=SlSideScreenPage
[Page_WidgetsPage]
ELEMENT100= name:=RMSElement,
implementation:=sleseasyscreen.SlEsSideScreenElement
```

- Create an element with the name "RMSElement" for displaying the Run MyScreens configuration.

**Note**

Use a number range ≥ 100 for new elements. This avoids conflicts with the SIEMENS display elements. The RMSElement is inserted in the WidgetsPage under the SIEMENS elements according to this numbering.

```
[Element_RMSElement]
PROPERTY001= name:=maskPath, type:=QString,
value:="sidescreenmask.com"
PROPERTY002= name:=maskName, type:=QString, value:="Mask"
```

- In the property maskPath, specify the name of the file that contains the RunMyScreens configuration, "sidescreenmask.com" in this case.
- In the Property maskName, specify the name of the RunMyScreens mask that is to be displayed, "Mask" in this case.

Save the file "sldmwidgets1.ini" in the directory */oem/sinumerik/hmi/cfg* or */user/sinumerik/hmi/cfg* .

The Run MyScreens configuration is available after the next HMI power-up.

## 12.4 Notes on carrying out Run MyScreens configurations in the Display Manager

Note the following:

- The functions LS, LM, DLGL, EXIT, EXITLS are not available when carrying out Run MyScreens configurations in the Display Manager.
- It is not possible to query the mask size in the LOAD block.
- Texts are always displayed using the font that is used in SINUMERIK Operate with a screen resolution of 640x480 pixels.
- The configured positions of operator controls are implemented without changes (e.g. extension).
- If Run MyScreens configurations are carried out in a side screen page, the entire area of the page is available for the configuration. When carrying out in side screen elements or widgets, the area available for display is specified by the system depending on the context.
- All of the debugging and error outputs are sequentially written to the text file easyscreen_log.txt. There is no specific identification indicating which configuration a message comes from.
- The status of a Run MyScreens configuration displayed in the Display Manager is rejected when the configuration is faded out.

**Note**

The integration of Run MyScreens configurations into the Display Manager can lead to more than one Run MyScreens configuration being carried out simultaneously. To maintain the functional capability of the overall system, take note of the additional system load caused as a result of the hardware used.

# Reference lists

# A

## A.1 Lists of start softkeys

### A.1.1 List of start softkeys for turning

#### Program operating area for turning

| HSK1 | HSK2 | HSK3 | HSK4 | HSK5 | HSK6 | HSK7 | HSK8 |
|---|---|---|---|---|---|---|---|
| **Edit** | **Drilling** | **Turning** | **Contour turning** | **Milling** | **Miscellaneous** | **Simulation** | **NC select** |
| HSK9 | HSK10 | HSK11 | HSK12 | HSK13 | HSK14 | HSK15 | HSK16 |
| -- | Straight-line circle (only for ShopTurn) | -- | -- | Measure workpiece | Measure tool | **OEM** | -- |

#### Turning

The following tables list the possible start softkeys for turning technology. Assignments of individual start softkeys can differ depending on the particular system. The specified OEM softkeys are permitted for "Run MyScreens".

programGUIDE (G-Code) start softkeys:

| | **Drilling** | **Turning** | **Contour turning** | | **Milling** | | **Miscellaneous** | |
|---|---|---|---|---|---|---|---|---|
| | **HSK2** | **HSK3** | **HSK4** | | **HSK5** | | **HSK6** | |
| **VSK1** | Centering | Stock removal | Contour | -- | Face milling | Contour | Settings | High speed settings |
| **VSK2** | Drilling reaming | Groove | Stock removal | -- | Pocket | Path | Swivel plane | Parallel axes |
| **VSK3** | Deep-hole drilling | Undercut | Stock removal residual material | -- | Multi-edge spigot | Predrilling | Swivel tool | -- |
| **VSK4** | Boring | Thread | Grooving | -- | Groove | Pocket | -- | -- |
| **VSK5** | Thread | Parting | Grooving residual material | -- | Thread milling | Pocket res. mat. | -- | -- |
| **VSK6** | **OEM** | -- | Plunge-turning | -- | Engraving | Spigot | Subprogram | -- |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **VSK7** | Positions | **OEM** | Plunge turning residual material | **OEM** | **OEM** | Spigot res. mat. | -- | **OEM** |
| **VSK8** | Repeat position. | -- | >> | << | Contour milling | << | >> | << |

ShopTurn start softkeys:

| | **Drilling** | **Turning** | **Contour turning** | | **Milling** | | **Miscellaneous** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **HSK2** | **HSK3** | **HSK4** | | **HSK5** | | **HSK6** | | **HSK10** |
| **VSK1** | Drilling centered | Stock removal | New contour | -- | Face milling | New contour | Settings | High speed settings | Tool |
| **VSK2** | Centering | Groove | Stock removal | -- | Pocket | Path | Swivel plane | Parallel axes | Straight line |
| **VSK3** | Drilling reaming | Undercut | Stock removal residual material | -- | Multi-edge spigot | Predrilling | Swivel tool | Repeat progr. | Circle center point |
| **VSK4** | Deep-hole drilling | Thread | Grooving | -- | Groove | Pocket | Counter-spindle | -- | Circle radius |
| **VSK5** | Thread | Parting | Grooving residual material | -- | Thread milling | Pocket res. mat. | Transformations | -- | Polar |
| **VSK6** | **OEM** | -- | Plunge-turning | -- | Engraving | Spigot | Subprogram | -- | Approach/retract |
| **VSK7** | Positions | **OEM** | Plunge turning residual material | **OEM** | **OEM** | Spigot res. mat. | -- | **OEM** | -- |
| **VSK8** | Repeat position. | -- | >> | << | Contour milling | << | >> | << | -- |

### See also

Defining the start softkey (Page 33)

## A.1.2 List of start softkeys for milling

### Program operating area when milling

| HSK1 | HSK2 | HSK3 | HSK4 | HSK5 | HSK6 | HSK7 | HSK8 |
|---|---|---|---|---|---|---|---|
| **Edit** | **Drilling** | **Milling** | **Contour milling** | **Turning** (only for G code) | **Miscellaneous** | **Simulation** | **NC select** |

| HSK9 | HSK10 | HSK11 | HSK12 | HSK13 | HSK14 | HSK15 | HSK16 |
|---|---|---|---|---|---|---|---|
| -- | Straight-line circle (only for ShopMill) | -- | -- | Measure workpiece | Measure tool | **OEM** | -- |

## Milling

The following tables list the possible start softkeys for milling technology. Assignments of individual start softkeys can differ depending on the particular system. The specified OEM softkeys are permitted for "Run MyScreens".

programGUIDE (G-Code) start softkeys:

| | **Drilling** | **Milling** | **Contour milling** | | **Turning** | | **Miscellaneous** | |
|---|---|---|---|---|---|---|---|---|
| | **HSK2** | **HSK3** | **HSK4** | | **HSK5** | | **HSK6** | |
| **VSK1** | Centering | Face milling | Contour | -- | Stock removal | Contour | Settings | -- |
| **VSK2** | Drilling reaming | Pocket | Path | -- | Groove | Stock removal | Swivel plane | Parallel axes |
| **VSK3** | Deep-hole drilling | Multi-edge spigot | Predrilling | -- | Undercut | Stock removal residual material | Swivel tool | -- |
| **VSK4** | Boring | Groove | Pocket | -- | Thread | Grooving | High speed settings | -- |
| **VSK5** | Thread | Thread milling | Pocket res. mat. | -- | Parting | Grooving residual material | -- | -- |
| **VSK6** | **OEM** | Engraving | Spigot | -- | -- | Plunge-turning | Subprogram | -- |
| **VSK7** | Positions | **OEM** | Spigot res. mat. | **OEM** | **OEM** | Plunge turning residual material | -- | **OEM** |
| **VSK8** | Repeat position. | -- | >> | << | Contour turning | << | >> | << |

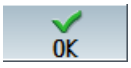
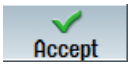
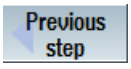
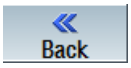
ShopMill start softkeys:

| | **Drilling** | **Milling** | **Contour milling** | | **Miscellaneous** | | **Straight line circle** |
|---|---|---|---|---|---|---|---|
| | **HSK2** | **HSK3** | **HSK4** | | **HSK6** | | **HSK10** |
| **VSK1** | Centering | Face milling | New contour | -- | Settings | -- | Tool |
| **VSK2** | Drilling reaming | Pocket | Path | -- | Swivel plane | Parallel axes | Straight line |
| **VSK3** | Deep-hole drilling | Multi-edge spigot | Predrilling | -- | Swivel tool | Repeat progr. | Circle center point |
| **VSK4** | Boring | Groove | Pocket | -- | High speed settings | -- | Circle radius |
| **VSK5** | Thread | Thread milling | Pocket res. mat. | -- | Transformations | -- | Helix |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **VSK6** | **OEM** | Engraving | Spigot | -- | Subprogram | -- | Polar |
| **VSK7** | Positions | **OEM** | Spigot res. mat. | **OEM** | -- | **OEM** | -- |
| **VSK8** | Repeat position. | -- | **>>** | **<<** | **>>** | **<<** | -- |

# A.2 List of predefined softkeys

Table A-1 Predefined softkeys

| Name | Softkey |
|---|---|
| SOFTKEY_OK |  |
| SOFTKEY_CANCEL |  |
| SOFTKEY_APPLY |  |
| SOFTKEY_MORE |  |
| SOFTKEY_BACK |  |
| SOFTKEY_ASSISTANT_NEXT |  |
| SOFTKEY_ASSISTANT_PREVIOUS |  |
| SOFTKEY_NAV_BACK |  |

# A.3 List of access levels

The meanings of the different access levels are as follows:

| Protection level | Locked by | Area |
|---|---|---|
| ac0 | Reserved for Siemens | |
| ac1 | Password | Machine manufacturer |
| ac2 | Password | Service |
| ac3 | Password | User |
| ac4 | Keyswitch position 3 | Programmer, machine setter |
| ac5 | Keyswitch position 2 | Qualified operator |
| ac6 | Keyswitch position 1 | Trained operator |
| ac7 | Keyswitch position 0 | Semi-skilled operator |

# A.4 List of colors

## System colors

A uniform color table is available for configuring dialogs (subset of the respective standard colors). The color of an element (text, input field, background, etc.) can be selected from the following options (between 0 and 133).

As an alternative to the predefined colors, you can also specify colors as RGB values ("#RRGGBB").

| Index | Pictogram | Color | Color description |
|---|---|---|---|
| 1 | | Black | |
| 2 | | Orange | |
| 3 | | Dark green | |
| 4 | | Light gray | |
| 5 | | Dark gray | |
| 6 | | Blue | |
| 7 | | Red | |
| 8 | | Brown | |
| 9 | | Yellow | |
| 10 | | White | |
| 126 | | Black | Font color of an input/output field that is currently in focus |
| 127 | | Light orange | Background color of an input/output field that is currently in focus |
| 128 | | Orange | System color focus |
| 129 | | Light gray | Background color |
| 130 | | Blue | Header color (active) |
| 131 | | Black | Header font color (active) |
| 132 | | Turquoise | Background color of a toggle field |
| 133 | | Light blue | Background color of a list box |

# A.5 List of language codes used in file names

## Supported languages

Standard languages:

| Language | Abbreviation in file name |
|---|---|
| Chinese simplified | chs |
| German | deu |
| English | eng |
| French | fra |
| Italian | ita |
| Spanish | esp |

Other languages:

| Language | Abbreviation in file name |
|---|---|
| Chinese traditional | cht |
| Danish | dan |
| Finnish | fin |
| Indonesian | ind |
| Japanese | jpn |
| Korean | kor |
| Malay | msl |
| Dutch | nld |
| Polish | plk |
| Portuguese (Brazil) | ptb |
| Romanian | rom |
| Russian | rus |
| Swedish | sve |
| Slovakian | sky |
| Slovenian | slv |
| Thai | tha |
| Czech | csy |
| Turkish | trk |
| Hungarian | hun |
| Vietnamese | vit |

# A.6 List of accessible system variables

## References

List Manual System Variables/PGAsl/

# A.7 Behavior when opening the dialog (attribute CB)

The following table provides an overview of the conditions when the CHANGE method is called.

The following applies to attribute CB:

**CB0**

The CHANGE method is triggered when the screen is displayed if the variable has a valid value at this time (e.g. through default setting or NC/PLC variable).

**CB1 (default)**

The CHANGE method is not explicitly triggered when the screen is displayed. If the variable has a configured NC/PLC variable, then the CHANGE method is of course still called.

| Condition | | | | Response |
|---|---|---|---|---|
| Type | System or user variable | Default setting | Execute the CHANGE method | |
| I/O field | Yes | Yes | Yes | Due to the configured NC/PLC variable, there is always at least one automatic call of the CHANGE method with the current value of the NC/PLC variable. The pre-assignment of CB does not have an effect. |
| | | | No | |
| | | No | Yes | |
| | | | No | |
| | No | Yes | Yes | The CHANGE method is called with the pre-assigned value. |
| | | | No | The CHANGE method is not called. |
| | | No | Yes | No call because there is no valid value present to call the CHANGE method. |
| | | | No | |
| Toggle | Yes | Yes | Yes | Due to the configured NC/PLC variable, there is always at least one automatic call of the CHANGE method with the current value of the NC/PLC variable. The pre-assignment of CB does not have an effect. |
| | | | No | |
| | | No | Yes | |
| | | | No | |
| | No | Yes | Yes | The CHANGE method is called with the pre-assigned value. |
| | | | No | The CHANGE method is not called. |
| | | No (per default the first value from the toggle is assigned) | Yes | The CHANGE method is called with the first pre-assigned value of the toggle list. |
| | | | No | The CHANGE method is not called. |

## See also

CHANGE (Page 127)

# Tips and tricks

# B

## B.1 General tips

- If possible, use the OPI version for system variables ($ variables).
  Reason:
  Avoids complex name resolution.
  Example:
  Instead of "$R[10]", it is better to use "/Channel/Parameter/R[u1,10]".
- Avoids cyclic (similar) setting, e.g. of the screen and variable HLP property. Compare the value to be set with the current value first, and only set it when it is different.
  Reason:
  Avoid a time-consuming search of the resolution-dependent help screens.
  Example:

```
DEF MyVar=(R3///,"X1",,"mm"/WR1//"$AA_IM[0]")
CHANGE(MyVar)
   IF MyVar.VAL < 100
      HLP="mypic1.png"
   ELSE
      HLP="mypic2.png"
   ENDIF
END_CHANGE
```

**Recommendation:**

```
CHANGE(MyVar)
   IF MyVar.VAL < 100
      IF HLP <> "mypic1.png"
         HLP="mypic1.png"
      ENDIF
   ELSE
      IF HLP <> "smypic2.png"
         HLP="mypic2.png"
      ENDIF
   ENDIF
END_CHANGE
```

- Replace several RNP() functions in succession with one MRNP() function.
  Replace several RDOP() functions in succession with one MRDOP() function.
  Reason:
  Reduces the communication load and increases the performance.
- Do not read drive parameters faster than in a 1 second cycle, slower is better.
  Reason: The communication with the drives can otherwise be extremely disturbed or even cause failures.
- Avoid successive arithmetic operations with system or user variable-connected dialog variables. Use, for example, register (REG[x]) or (invisible) help variables for this.
  Reason: Each assignment of a value results in a write operation in the connected system or user variable.
- Similar code used in different blocks should be combined in a SUB block for reasons of transparency, serviceability and performance (when displaying the screen). It can then be called at the appropriate positions with the CALL() function.
- By monitoring the CPU load in the dialog line (setting in slguiconfig.xml), it is possible to investigate the effects of changes in the screen on the performance.

## B.2 Tips for debugging

- Use the DEBUG() function for diagnostic purposes. When the DEBUG() function is called, the transferred string is written to the "easyscreen_log.txt" file. The output of information in the dialog line by the DLGL() function can also be helpful.
  However, after completion of the screen development, this function should be removed again for performance reasons or commented out.
- The "easyscreen_log.txt" file should have no entries after the development of a screen is completed.

## B.3 Tips for the CHANGE method

- Always keep CHANGE methods very short and small, particularly for those whose variables are connected to a system or user variable and change very frequently.
  Reason:
  Increases the screen performance.
- If possible, do not configure any RNP() functions in CHANGE methods. Instead, it is better to configure an invisible variable parallel to the system or user variable to be read and then use this.
  Reason:
  An RNP() function would be issued with each call. Otherwise, the current value which is already available would be simply accessed.

Example:

A name resolution is performed via RNP() for each change of the axis motion in order to read channel-specific machine data:

```
DEF AXIS_POSITION_X =  (R///,""///"$AA_IM[X]")

CHANGE(AXIS_POSITION_X)
      DLGL("Axis """ << RNP("$MC_AXCONF_GEOAX_NAME_TAB[0]") << """
has moved: "
<< AXIS_POSITION_X)
END_CHANGE
```

Keep the channel-specific machine data up-to-date with the aid of an invisible variable, copy each value change to a temporary variable, e.g. register.

This temporary variable can then be used in the CHANGE method of the value change for the axis position without making a name resolution of the machine data each time and the subsequent read access:

```
DEF AXIS_POSITION_X =  (R///,""///"$AA_IM[X]")
DEF AXIS_NAME_X = (S///,""/WR0//"$MC_AXCONF_GEOAX_NAME_TAB[0]")

CHANGE(AXIS_NAME_X)
      REG[0] = AXIS_NAME_X
END_CHANGE

CHANGE(AXIS_POSITION_X1)
   DLGL("Axis """ << REG[0] << """ has moved " << AXIS_POSITION_X)
END_CHANGE
```

- The update rate and therefore the execution of the associated CHANGE method of variables that are connected to system or user variables with very frequent value changes, can be reduced by using the UR variable property, e.g. variable that is coupled to the actual axis values.
  Reason:
  In this way, the associated CHANGE method is executed in a fixed specified grid for a value change.

## B.4 Tips for DO LOOP loops

As loops can impair the performance of SINUMERIK Operate, depending on the configuration, they should be used carefully and if possible without time-intensive actions.

A register (REG[]) should also be used as run variable, because normal display variables (particularly those with OPI connection) also impair the system performance due to the frequent updates or write operations.

Note that the number of script lines executed "in one operation", is currently limited to 10,000. When this number is reached, the script execution is aborted with an appropriate entry in the "easyscreen_log.txt" file.

Reason:

- Prevention of endless loops
- "Run MyScreens" must remain operable

# Animated elements

C

## C.1 Introduction

### Introduction

The manual describes working with the X3D Viewer with which you can integrate animated and non-animated graphical scenes (animated elements) – called help screens in the following – into the graphical user interface of SINUMERIK Operate as of Version V4.7 SP1 .

You can show motion sequences and parameters in context-sensitive help screens. This helps you to improve the operation of applications and make the user interface more appealing.

Implementation of your initial idea to the final help screen is described in the steps below:

- Creation of graphical elements and 3D models for the help screens that will eventually appear in the X3D Viewer (see Chapter Modeling (Page 321)).
- Creation of the scene description file in which the model data of the graphics file will be assigned to the scenes and animations to be displayed at particular points (see Chapter Structure of the scene description file (Page 327)).
- Conversion of the X3D files and XML files into HMI files (see Chapter Conversion to hmi file (Page 332)).
- C++ integration of the X3D Viewer into your own application (see Chapter Implementation example (Page 335)).
- Application in Run MyScreens (see Chapter Display in Run MyScreens (Page 336)).

### Overview of the file formats

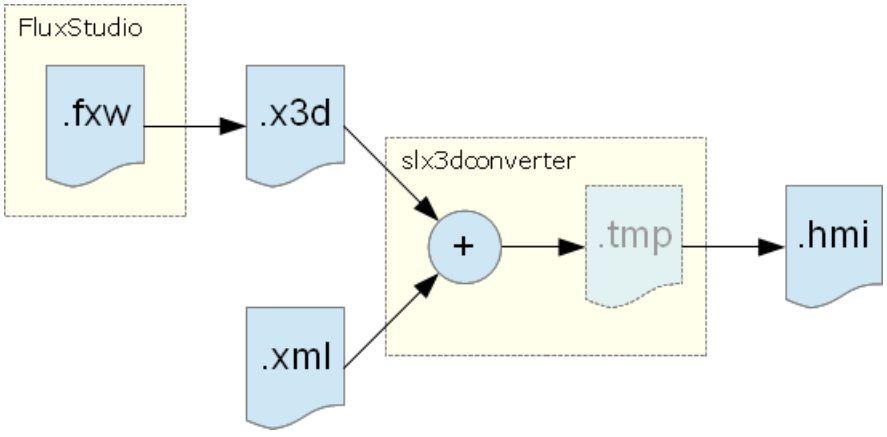



Figure C-1 File formats

| File format | Description |
|---|---|
| .fxw | Flux Studio (3D) graphics file |
| .x3d | Model file in exchange format |
| .xml | Definition file for animations and scenes |
| .hmi | Output file for the X3D Viewer |

# C.2 Modeling

## C.2.1 Requirements

The aim of the modeling is to create a file with the generated 3D models in .x3d file format, an official standard for 3D contents.

Modeling is performed in the **Flux Studio Web3D authoring tool** . The Flux Studio Web3D authoring tool is a freely available 3D modeling tool with versatile export and import options.

### Requirements

- You have installed the Flux Studio Web3D authoring tool, Version 2.1.
- You are familiar with modeling and use of Flux Studio.
- File format .x3d is not described in any detail in this document, adequate knowledge of the subject is assumed.

**Note**

You can download the Flux Studio Web3D authoring tool from the Internet via the following link:

http://mediamachines.wordpress.com/flux-player-and-flux-studio (http://mediamachines.wordpress.com/flux-player-and-flux-studio)

## C.2.2 Rules for modeling

Observe the following rules when modeling. You will only be able to process your help screen file if you have followed these rules correctly.

### Rules for modeling

1. The time sensor must have the name "Animation".
2. All related elements must be assigned to a group.
3. All groups must be set to the following position:
   x = 0, y = 0, z = 0
4. To make the groups invisible, the translation values are set as follows:
   x = 1000000, y = 1000000 and z = 1000000
5. The following elements must have the same name both in the Flux Studio graphic models and in the XML scene definitions (cf. Chapter Overview (Page 327)):
   - Tool
   - Cameras

6. Before you can create a .x3d file you must make the following settings in dialog box "Export Options":

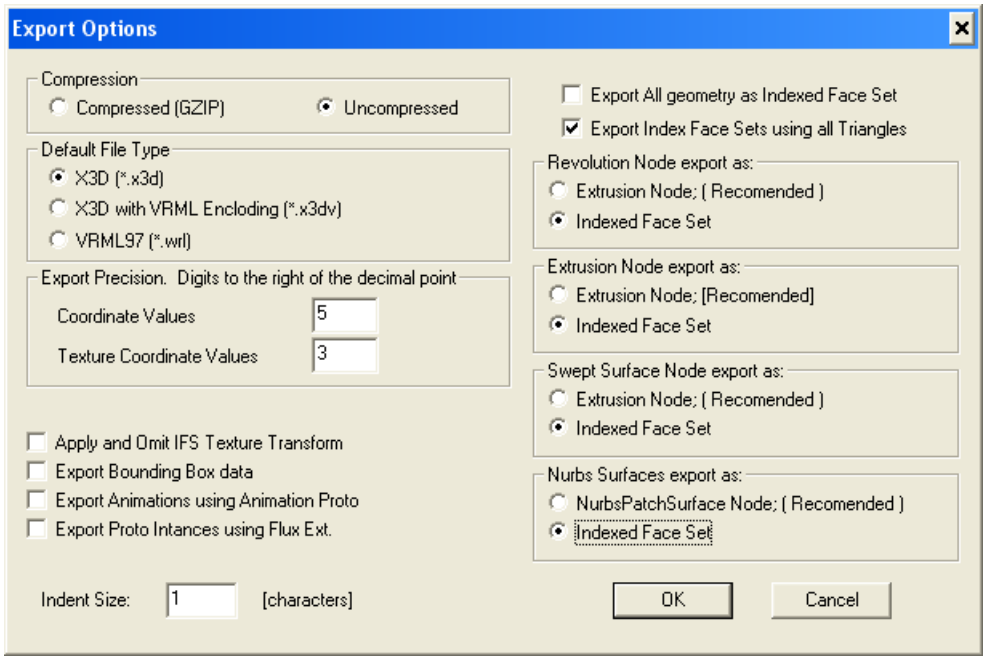



Figure C-2 Settings in the Export Options dialog box

7. We recommend the following settings for texts (see also the following dialog box):
   - Texts must always be centered.
   - The size of the text should be 0.2. This size allows you to position the text easily. The text output by the X3D Viewer is in the font size of the user interface, irrespective of this value.

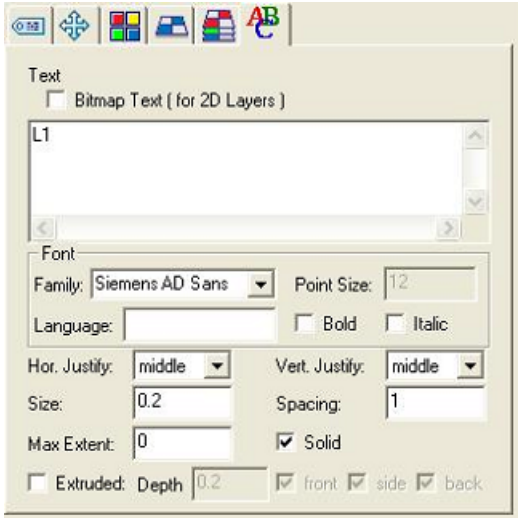



Figure C-3 Settings for text

8. To create hatching, use file schnitt.png as the texture. The lines always go from the bottom left to the top right at an angle of 45°. You should set the scaling to 3 in both dimensions. The rotation depends on the construction type and must be adapted manually.

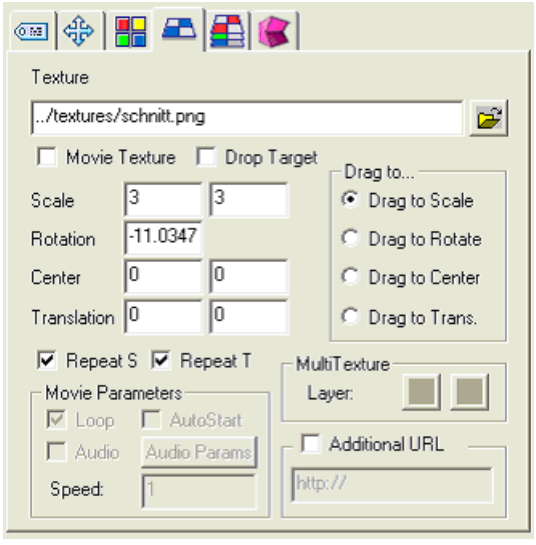



Figure C-4 Settings for hatching

9. We recommend the following sizes/dimensions for the blanks:
   - Cylinder for turning operation:
     Length = 5.5 ; Radius = 1.4 (cf. template *turning_blank.fxw*)
   - Cube for milling operation
     x = 4.75 ; y = 3 ; z = 3 (cf. template *milling_blank.fxw*)

### See also

XML commands (Page 327)

## C.2.3 Importing graphics (models)

In Flux Studio, you can import external graphics (models). You can store frequently used models, such as tools, as .x3d or .hmi files centrally for further use as finished elements in other help screens. You will find a list of supplied modeling templates in Chapter Modeling templates (Page 325).

Complex objects can also be created with other modeling tools, such as Cinema4D, and then imported.

A description of how you can reuse these models is given below.

### Import as inline

Flux Studio uses the object "Inline" for importing .x3d files. It allows you to link in external elements without multiplying the 3D data of these models.

Insert the object with menu item **Create** -> **Create Inline**. Then enter a file name in the object properties.

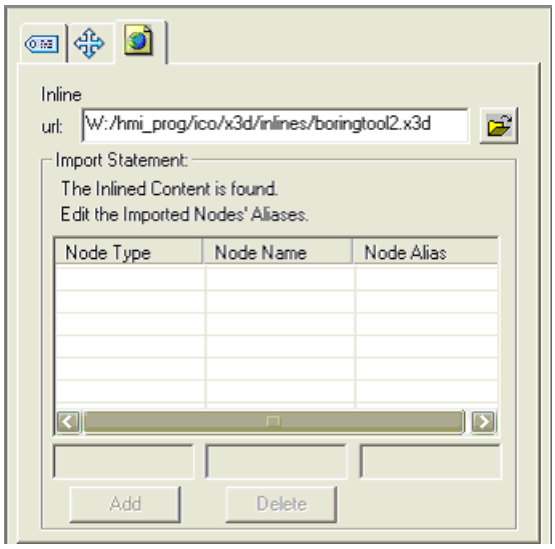



Figure C-5 Importing with object "Inline"

### Importing 3D model data

Proceed as follows to import model data from a file.

1. Open dialog box "Flux Studio Accutrans3D Translation Utility" via menu item **File** -> **Import Other Format** .
2. Select a format from the "File Types" list box. For example, to import a Cinema4D file, select file type "3D Studio."
3. Then select a file with **Select Files** and start import.

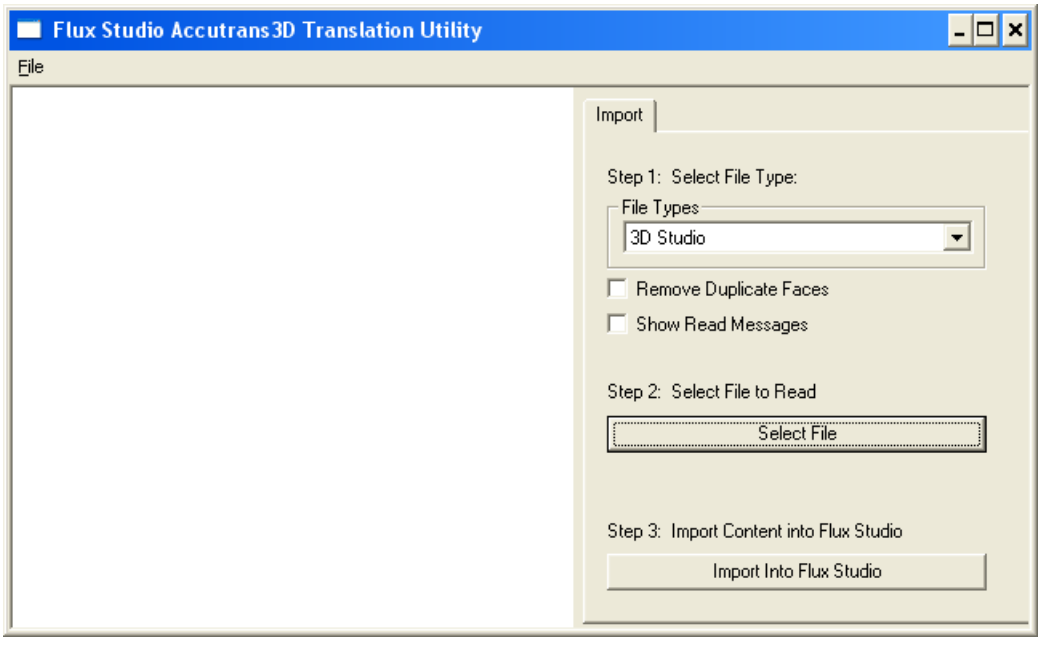



Figure C-6 Importing 3D model data

The model data from the Cinema4D file is inserted as a complete group. All cameras and any other information that is additionally imported should be deleted. Only the graphics elements of the Cinema4D file are required.

## C.2.4 Modeling templates

In this chapter, you will find a list of graphics elements on which you can base your design, colors, and sizes. To maintain a uniform "look and feel" in the user interface, we recommend using the style of the templates or the templates themselves for creating your own graphics.

### General

| Template | Description |
|---|---|
| intersection_texture.png | Texture for an sectioned area |
| rapid_traverse_line_hori.fxw | Horizontal line to illustrate the tool traversing path during rapid traverse |
| rapid_traverse_line_vert.fxw | Vertical line to illustrate the tool traversing path during rapid traverse |

| Template | Description |
|---|---|
| feed_traverse_line_hori.fxw | Horizontal line to illustrate the tool path during feed |
| feed_traverse_line_vert.fxw | Vertical line to illustrate the tool path during feed |
| dimensioning_lines_hori.fxw | Horizontal projection lines |
| dimensioning_lines_vert.fxw | Vertical projection lines |
| z1_inc.fxw | Horizontal dimension line with the identifier Z1 |
| x1_inc.fxw | Vertical dimension line with the identifier X1 |
| 3d_coordinate_origin.fxw | 3D coordinate origin |
| 3d_zero_point.fxw | 3D zero point |

## Turning

| Template | Description |
|---|---|
| turning_blank.fxw | The blank for turning technology: A simple cylinder |
| turning_centerline.fxw | Center line |
| turning_centerpoint.fxw | Coordinate origin to display the zero point in the WCS |
| turning_refpoint.fxw | Reference point, for example, for a machining operation |
| turning_machining_area.fxw | Boundary lines for the machining area |

## Milling

| Template | Description |
|---|---|
| milling_blank.fxw | The blank for milling technology: A simple cube |
| milling_centerline.fxw | Center line |
| milling_refpoint.fxw | Reference point, for example, for a machining operation |

# C.3 XML commands

## C.3.1 Overview

The scene description file (*.xml) is required so that the X3D Viewer can access the graphics. In the scene description file, you assign the scene names that are called from the configuration to the times in the *.x3d file (time sensor) at which the graphics are positioned.

## C.3.2 Structure of the scene description file

The structure of the scene description file is shown below together with an explanation of the relevant XML commands:

### Design and description

**<!-- Treatment of the plane-dependent texts for turning -->**

```
<TextPlane plane="G18_ZXY" />
```

**Description**

| | |
|---|---|
| <TextPlane plane="G18_ZXY" /> | Treatment of the plane-dependent texts (axis names) specifically for turning. If, for example, a text with the identifier "Z" is used (for G18 first geometry axis), the corresponding geometry axis identifier of the control will be shown in the help screen (for example, "Z"). |

**<!--Text alignment -->**

```
<TextPosition center='true' />
```

**Description**

| | |
|---|---|
| <TextPosition center='true' /> | Marking that the texts are centered. This is relevant to the display of the texts in rotated screens. We recommend using this setting. |

**<!--Adapting the tool speed -->**

```
<ToolSpeedFactors planeSpeedFactor="0.4" rapidSpeedFactor="0.7"
reducedSpeedFac¬tor="1.0"/>
```

### Description

| | |
|---|---|
| <ToolSpeedFactors | This entry can be added if the tool speeds are to be modified. |
| planeSpeedFactor="0.4" | Factor for the feedrate ( 0.4 = 40% ) |
| rapidSpeedFactor="0.7" | Factor for rapid traverse ( 0.7 = 70% ) |
| reducedSpeedFactor="0.1" /> | Factor for a third speed ( 0.1 = 10% ) |

**<!--Definition of an animation-->**

```
<SceneKey name='BoringAnimation' masterRotationSpeed="-64.0"
maxRotAngle="45.0" begin-Time='0.110 endTime='0.118 view='camiso'
speedMaster='boringtool' Type="VIEW_3D_TURN_CYL">
```

### Description

| | |
|---|---|
| <SceneKey name='BoringAnimation' | Name of the animation |
| masterRotationSpeed="-64.0" | Direction of rotation and rotation speed of the tool.<br>The setting is optional. |
| maxRotAngle="45.0" | To avoid aliasing effects (tool seems to be turning in the wrong direction). The value is usually a fourth of the symmetry of the tool.<br>The setting is optional. |
| beginTime='0.110 | Start time of the animation on the time sensor |
| endTime='0.118' | End time of the animation on the time sensor |
| view='camiso' | Camera to be used for the animation |
| speedMaster='boringtool' | Name of the tool for the animation |
| type="VIEW_3D_TURN_CYL"> | View type defines the view (see Section View type (Page 330)) |

**<!-- Elements -->**

```
<Element name='1' time='0.110' feed='rapid' dwell='2'/>
<Element name='2' time='0.112' feed='rapid'/>
<Element name='3' time='0.114' feed='rapid'/>
<Element name='4' time='0.116' feed='plane'/>
<Element name='5' time='0.118' feed='plane'/>
```

### Description

| | |
|---|---|
| <Element name='1' | Specifies that it is the first element of the animation. |
| time='0.110' | Time of the first element on the time sensor |

| | |
|---|---|
| feed='rapid' | Traversing speed of the element<br>plane = machining feedrate<br>rapid = rapid traverse<br>reduced = reduced speed, slower than "plane" |
| dwell=’2’/> | Pause in the traversing motion of the animation. A rotating tool continues to rotate. |

**<!-- End of animation definition -->**

```
</SceneKey>
```

**Description**

| | |
|---|---|
| </SceneKey> | End of the animation definition |

**<!-- Existing animation as source for a new animation -->**

```
<SceneKey source=''BoringAnimation' name=''BoringAnimationRear' mirror='
MirrorScreenX ' />
```

**Description**

| | |
|---|---|
| <SceneKey source="BoringAnimation' | Name of an animation that is to use another animation as its source. |
| name="BoringAnimationRear' | Name of the new animation based on the source |
| mirror=' MirrorScreenX ' /> | Mirroring in the direction of the X axis, output is in the new animation. |

**<!-- Definition of a static scene (screen), (four examples) -->**

```
<SceneKey name='Default' time='0.026' view='camiso'
type="VIEW_3D_DRILL_CUT"/>
<SceneKey name='Cut' time='0.046' view='camside' type="VIEW_SIDE"/>
<SceneKey name='Z1ink' time='0.056' view='camside' type="VIEW_SIDE"
 highLightedGroup='dad_Z1ink'/>
<SceneKey name='Z1abs' time='0.066' view='camside' type="VIEW_SIDE"
 highLightedGroup='dad_Z1abs'/>
```

**Description**

| | |
|---|---|
| <SceneKeyname=’Z1abs' | Screen name |
| time='0.066' | Time of the screen on the time sensor |
| view='camside' | Camera to be used for the screen |
| type="VIEW_SIDE" | View type defines the view (see Section View type (Page 330)) |
| highLightedGroup='dad_Z1abs'/> | Name of the group to be highlighted. The group has been called "Z1abs" in Flux Studio. The prefix "dad_" is automatically added by Flux. |

```
<!– End of xml file –!>
```

## C.3.3 Mirroring and rotations

Mirroring or rotation of the image or model is defined with the **mirror** command.

### Description

| | |
|---|---|
| mirror="RotateScreenX" | The screen is rotated around the X axis |
| mirror="RotateScreenY" | The screen is rotated around the Y axis |
| mirror="RotateScreenZ" | The screen is rotated around the Z axis |
| mirror="MirrorScreenX” | The screen is mirrored along the X axis |
| mirror="MirrorScreenY" | The screen is mirrored along the Y axis |
| mirror="MirrorScreenZ" | The screen is mirrored along the Z axis |
| mirror="RotatePieceX” | The model is rotated around the X axis |
| mirror="RotatePieceY" | The model is rotated around the Y axis |
| mirror="RotatePieceZ" | The model is rotated around the Z axis |
| mirror="MirrorPieceX" | The model is mirrored along the X axis |
| mirror="MirrorPieceY" | The model is mirrored along the Y axis |
| mirror="MirrorPieceZ" | The model is mirrored along the Z axis |

All rotate and mirror options can be combined. A rotation angle must be defined for rotation.

### Examples

mirror="RotatePieceZ=180"

mirror="RotatePieceZ=-90 MirrorScreenX"

mirror="RotateScreenY=90"

mirror="MirrorPieceX MirrorPieceY"

mirror="MirrorPieceZ RotatePieceX=-90"

## C.3.4 View type

You define the views with the view type. The views are based on empirical values and rules that produce a sensible representation of the objects.
This ensures that the representation of the objects matches the coordinate system settings made in the user interface (MD 52000 $MCS_DISP_COORDINATE_SYSTEM or MD 52001 $MCS_DISP_COORDINATE_SYSTEM_2).

## Description

| | |
|---|---|
| "VIEW_STATIC" | Direct view without conversion |
| "VIEW_3D_TURN_CYL" | 3D view for turning operations (cylinder) |
| "VIEW_3D_MILL_CUBE" | 3D view for milling operations (cube) |
| "VIEW_3D_DRILL_CUT" | 3D view for drilling operations (sectional view) |
| "VIEW_SIDE" | 2D view for drilling operations (sectional view) |
| "VIEW_TOP_GEO_AX_1" | 2D view from the direction of the 1st geometry axis |
| "VIEW_TOP_GEO_AX_2" | 2D view from the direction of the 2nd geometry axis |
| "VIEW_TOP_GEO_AX_3" | 2D view from the direction of the 3rd geometry axis |

# C.4 Conversion to hmi file

**Note**

The X3D files and their associated XML files are converted to HMI files when the HMI is powered up.

For each X3D file, a corresponding XML file of the same name must be created. To do this, you store the X3D files and XML files in directory `search path of HMI\ico\x3d \turning` or `milling`. You proceed in the same way with the .ts files.

# C.5 Display in Create MyHMI /3GL

## C.5.1 X3D Viewer

If you want to display particular help screens in a separate OA application you must integrate the X3D Viewer widget into its own OA application.

The X3D Viewer widget provides interfaces, which enable X3D contents to be presented in the HMI.

Class SIX3dViewerWidget is provided for displaying graphical scenes. The definition of the class can be found in the corresponding slx3dviewerwidget.h header file in the global GUI include directory `\hmi_prog\gui\include`.

## C.5.2 Class SIX3dViewerWidget

The class provides a flexible widget that autonomously displays the contents of a model file specified during runtime and, if required, runs the animation.

The interface of the class comprises the Constructor, Destructor, and two methods for controlling the graphical output.

As a direct derivative of the Qt class QWidget, a much more expansive interface is available, which will not be described in any further detail here, (for example, show(), hide() and resize(...). For more information, please refer to the Qt documentation).

## C.5.3 Public methods

### SIX3dViewerWidget ( QWidget* pParent = 0 )

Constructor of the X3D Viewer widget.

| Parameters | Meaning |
|---|---|
| pParent | The parameter is passed on to the constructor of the Qwidget. |

### ~SIX3dViewerWidget ( )

Destructor of the X3D Viewer widget.

| Parameters | Meaning |
|---|---|
| - | - |

## C.5.4 Public slots

### void viewSceneSlot ( const QString& rsFileName, const QString& rsScene, const QString& rsAnimationScene, int nChannel, int nPlane, SlStepTechnology nTechnology )

With the viewSceneSlot method, the X3D Viewer is instructed to load the rsScene static scene and the rsAnimationSzene animated scene from the rsFileName file and display them alternately.

The static scene and then the animated scene are repeatedly alternately displayed for a fixed time.

If no static scene is specified, the animation will be shown immediately; an animated scene may also not have been specified.

Channel number, plane and technology are used to rotate the scenes to the correct position (depending on the set machine coordinate system).

| Parameters | Meaning |
|---|---|
| rsFileName | Name of the file that contains the scenes to be displayed |
| rsScene | Name of the static scene. |
| rsAnimationScene | Name of the animated scene. |
| nChannel | Channel number |
| nPlane | Plane |
| nTechnology | Technology<br>The following constants are defined for the SlStep-Technology enumeration type:<br>• SL_STEP_NO_TECHNOLOGY<br>• SL_STEP_MILLING<br>• SL_STEP_TURNING<br>• SL_STEP_SURFACE_GRINDING<br>• SL_STEP_CIRCULAR_GRINDING |

### void viewSceneSlot ( const QString& rsFileName, const QString& rsScene, const QString& rsAnimationScene )

A simplified form of the viewSceneSlot method with which the X3D Viewer can be instructed to load and display the rsScene static scene and/or the rsAnimationSzene animated scene from the rsFileName file.

| Parameters | Meaning |
|---|---|
| rsFileName | Name of the file that contains the scenes to be displayed. |
| rsScene | Name of the static scene. |
| rsAnimationScene | Name of the animated scene. |

## C.5.5 Libraries

In order to be able to apply the X3D Viewer in your own projects, the list of library dependencies must be extended by the 'slx3dviewer.lib' entry.

## C.5.6 Implementation example

You will find an implementation example in the package Create MyHMI/3GL under **\examples \GUIFrameWork\SlExGuiX3D**.

## C.5.7 Machine data

Display MD 9104 $MM_ANIMATION_TIME_DELAY

Time delay in seconds until the animation is started for help screens. The setting does not apply to help screens that are only animated.

The setting applies globally to all animations of SINUMERIK Operate.

## C.5.8 Notes about use

- The animation is interrupted if the X3D Viewer widget is hidden. If this happens, there is no need to choose a scene selection without animation.
- You should avoid frequent or multiple instantiation of the X3D Viewer widget as this reduces performance and memory capacity. The use (implementation) of an X3D Viewer Singleton is recommended in such application scenarios.
- The X3D Viewer can also be used in signal slot concepts.
- If an error occurs (for example, a file not found or unknown scene name), the X3D Viewer widget opens a message area. This area is automatically hidden again when another help screen is activated.

# C.6 Display in Run MyScreens

This section is intended for experienced "Run MyScreens" developers. Necessary background knowledge is provided in the associated documentation.

In addition to using screens in .bmp or .png format, animated help screens can also be displayed with the X3D Viewer.

To integrate help screens, use the Run MyScreens interface in the usual way.

To output images in .bmp or .png format, XG0 is entered in the definition of a dialog box in the Attributes section or the parameter is left empty. To integrate X3D help screens in a dialog, you must enter **XG1** in the attributes.

```
//M(MASK_F_DR_O1_MAIN/$85407////52,80/XG1)
```

The following parameters are required to activate the individual help screens; their meaning is described in Section 5.3 Public slots:

1. File name
2. StaticScene (optional)
3. AnimationScene (optional)
4. Technology (optional)
5. Plane (optional)

The parameters are combined into a character string in this order, separated by a comma.

```
Hlp = "File name,StaticScene,AnimationScene,Technology,Plane"
```

## Examples

- The default help screen can be set with the specified variable Hlp.
  In the following example, the "MyAnimation" animation is output from the "MyDlgHelp.hmi" file. No StaticScene is specified, i.e. no static scene is output. No specifications are made for the Technology and Plane, i. e. default values are used.
  ```
  Hlp = "MyDlgHelp.hmi,,MyAnimation"
  ```
- The hlp property can be set on a specific help screen for the individual parameters of the input screen.
  In the following example, the static "MyParam" scene and the "MyAnimation" animation are output from the "MyDlgHelp.hmi" file in the G17 plane. No specification is made for the Technology, i.e. a default value is used.
  ```
  VarMyParam.hlp = "MyDlgHelp.hmi,MyParam,MyAnimation,,G17"
  ```

# Glossary

## Access level

Graduated system of authorization, which makes the accessibility and utilization of functions on the operator interface dependent on the authorization rights of the user.

## Array

An array can be used to organize data of a standard data type stored in the memory in such a way that it is possible to access the data via an index.

## Attribute

Characteristic that assigns specific → Properties to an object (→ Dialog or → Variable).

## Column index

Column number of an array

## Configuration file

File, which contains definitions and instructions that determine the appearance of → Dialogs and their → Functions.

## Definition lines

Program section in which → Variables and softkeys are defined

## Dialog

Display of the → User interface

- **Dialog-dependent softkey menu**
  Softkey menu, which is called from a newly configured dialog.
- **Dialog-independent softkeys**
  Softkeys, which are not called from a dialog, i.e., start softkey and softkey menus, which the user configures before the first, new dialog.

## Editor

ASCII Editor with which characters can be entered in a file and edited.

### Event

Any action, which initiates execution of a → Method: Input of characters, actuation of softkeys, etc.

### Group

Reload unit for → Configuration file

### Help variable

Internal arithmetic variable to which no → Properties can be assigned and is not, therefore, visible in the → Dialog.

### Hotkeys

6 keys on OP 010, OP 010C and SINUMERIK keyboards with hotkey blocks. Pressing the keys selects an operating area directly. As an option, 2 additional keys can be configured as hotkeys.

### Input/output field

Also I/O field: for inputting or outputting variable values.

### Interpreter

The interpreter automatically converts the defined code from the → Configuration file into a → Dialog and controls its use.

### Line index

Row number of an array

### Menu tree

A group of interlinked → Dialogs

### Method

Programmed sequence of operations executed when a corresponding → Event occurs.

### Parameter

Parameters are variable elements of the programming syntax and are replaced by other words/ symbols in the → Configuration file.

### PI service

Function which, on an NC, executes a clearly defined operation. PI services can be called from the PLC and the HMI system.

### PLC hard key

PLC hard keys are provided via the PLC interface of the HMI software, just like hotkeys. The functions, which can be triggered via the user interface, can be configured.

They take the form of MCP keys or evaluations of PLC signal logic operations in the PLC user program. For this reason, they are referred to as "virtual keys".

### Programming support

Provision of → Dialogs to assist programmers in writing → Parts programs with "higher-level" components

### Properties

Characteristics of an object (e.g of a → Variable)

### Recompile

NC code sections can be generated in a → Part program from input fields in → Dialogs in the → Programming support system. Recompilation is the reverse operation. The input fields used to generate a selected section of NC code are retrieved from the NC code and displayed in the original dialog.

### Selecting

A program formulated in the NC language, which specifies motion sequences for axes and various special actions.

### Simulation

Simulation of a → Parts program run without movement of the actual machine axes.

### Softkey labels

Text/image on the screen, which is assigned to a softkey.

### Softkey menu

All horizontal or all vertical softkeys

### Start softkey

Softkey with which the first newly created → Dialog is started.

### Toggle field

A list of values in the → Input/output field; check with toggle field: The value input in a field must be the same as one of the listed values.

### User variable

Variables defined by the user in the → Parts program or data block.

### Variable

Designation of a memory location, which can be displayed in a → Dialog by assigning → Properties and in which input data and the results of arithmetic operations can be entered.

# Index


## A

Access level, 63
Alarms
  Language code, 310
Array
  Access mode, 197
  Column index, 197
  Compare mode, 197
  Definition, 196
  Element, 197
  Line index, 197
  Status, 200
Attributes, 93

## B

Background color, 95

## C

Colors, 95
Comparison operators, 123
Conditions, 124
Configuration file, 29
Configuration syntax"; "Extended syntax, 43
Configuration syntax"; "Screens, 43
Configuration syntax"; "Softkey, 44
Configuration syntax"; "Table columns, 44
Configuration syntax"; "Variables, 43
Configuring PLC softkeys, 281
Constants, 123
Custom widget
  Automatic data exchange, 209
  Definition, 206
  Executing a method, 212
  Interface, 207
  Library, 206
  Manual data exchange, 210
  Reading and writing properties, 210
  Responding to a custom widget signal, 214

## D

Default setting, 92
Defines the softkey menu, 62
Dialog
  Definition, 47
  Definition block, 48
  Multiple columns, 58
  Properties, 49
Dialog change mode, 165
Dialog element, 56
Display mode, 93
Display option, 93
DLL file, 157

## E

ELLIPSE (define circle), 194
ELLIPSE (define ellipse), 194

## F

File
  Copy, 144
  Delete, 144
  Moving, 146
File formats
  fxw, 319
  hmi, 319
  x3d, 319
  xml, 319
Flux Studio Web3D authoring tool, 321
Focus control, 205
Foreground color, 95
Function
  CALL (Subprogram call), 140
  CLEAR_BACKGROUND), 143
  CP (Copy Program), 144
  CVAR (Check Variable), 142
  Cyclic execution of scripts, 190
  DEBUG, 151
  DLGL (Dialog line), 150
  DO LOOP), 188
  DP (Delete Program), 144
  EP (Exist Program), 145
  EVAL (Evaluate), 156
  EXIT, 152
  EXITLS (EXIT Loading Softkey), 157
  FCT, 157
  File access, 148
  FORMAT (string), 187
  GC (Generate Code), 159

HMI_LOGIN, 162
HMI_LOGOFF, 162
HMI_SETPASSWD, 162
INSTR (String), 182
LA (Load Array), 163
LB (Load Block), 164
LEFT (string), 183
LEN (string), 182
LISTADDITEM), 154
LISTCLEAR), 154
LISTCOUNT), 154
LISTDELETEITEM, 154
LISTINSERTITEM, 154
LM (Load Mask), 165
LS (Load Softkey), 166
MIDS (string), 184
MP (Move Program), 146
MRDOP), 139
MRNP (Multiple Read NC PLC), 169
Overview, 138
PI_START, 171
RDFILE), 148
RDLINEFILE), 148
RDOP), 139
Reading and writing drive parameters, 139
Recompile NC code, 176
Recompile without comment, 178
REPLACE (string), 184
RESIZE_VAR_IO, 173
RESIZE_VAR_TXT, 173
RETURN (Back), 175
RIGHT (string), 183
RNP (Read NC PLC Variable), 172
SB (Search Backward), 181
SF (Search Forward), 181
SP (Select Program), 147
START_TIMER), 190
STOP_TIMER, 190
STRCMP (string), 185
STRINSERT (string), 185
STRREMOVE (string), 186
SWITCH), 169
TRIMLEFT (string), 186
TRIMRIGHT (string), 186
UNTIL, 188
WDOP, 139
WHILE, 188
WNP (Write NC PLC Variable), 172
WRFILE, 148
WRLINEFILE), 148

## G

Generates NC code, 159
Graphic text, 92
Grid → Table, 202

## H

H_SEPARATOR (define horizontal dividing line), 195
Help display, 95
Help file, 96
Help variable, 82

## I

I/O field with integrated unit selection, 96
Image as short text, 88
Import
  Graphics (models), 323
Input mode, 94

## L

Language code, 310
Language-dependent texts, (SIEsTouchButton)
Limits, 92
LINE (define line), 193
List field, 87
Long text, 92

## M

Machine data, 335
Machining step support, 178
Master dialog, 165
Mathematical functions, 122
Menu tree, 31
Method
  ACCESSLEVEL, 126
  CHANGE, 127
  CHANNEL, 128
  CONTROL, 129
  LANGUAGE, 130
  LOAD, 130
  LOAD GRID, 167
  OUTPUT, 132
  Overview, 126
  PRESS, 133

PRESS(ENTER), 134
PRESS(TOGGLE), 134
RESOLUTION, 135
RESUME, 135
SUSPEND, 136
UNLOAD, 131
Multitouch operation, 247

## N

NC variable
Read, 172
Write, 172
Numerical format, 100

## O

Operator
Bit, 124
Mathematical, 121

## P

Password dialogs, 59
Password input mode (asterisk), 91
PI services, 138
PLC variable
Read, 172
Write, 172
Position
Input/output field, 95, 104
Short text, 95, 104
Progress bar with two color changes, 89
Progress bar without color change, 90
Protection levels, 308
Public slots, 334

## R

RECT (defining a rectangle), 193
Registers
Exchanging data, 174
Status, 175
Value, 174

## S

Short text, 92
SIEsButton
Overview of properties, 249
SIEsGraphCustomWidget
Reading and writing properties, 219
SIEsGraphCustomWidgets
addArc, 240
addCircle, 239
addContour, 233
addEllipse, 239
addLine, 238
addPoint, 238
addRect, 238
addRoundedRect, 239
addText, 240
AxisNameX, 220
AxisNameY, 221
AxisY2Factor, 221
AxisY2Offset, 221
AxisY2Visible, 221
BackColor, 227
ChartBackColor, 227
clearContour, 235
CursorColor, 228
CursorStyle, 230
CursorX, 229
CursorY, 229
CursorY2, 229
findX, 236
fitViewToContour, 235
fitViewToContours, 235
ForeColor, 227
ForeColorY2, 227
GridColor, 228
GridColorY2, 228
hideAllContours, 234
hideContour, 234
KeepAspectRatio, 225
moveCursorOnContourBack, 244
moveCursorOnContourBegin, 243
moveCursorOnContourEnd, 243
moveCursorOnContourNext, 244
Overview of functions, 231
Overview of properties, 220
Overview of signals, 245
removeContour, 234
repaint, update, 237
ScaleTextEmbedded, 223
ScaleTextOrientationYAxis, 224
SelectedContour, 226
serialize, 245, 266
setCursorOnContour, 242
setCursorPosition, 241
setCursorPositionY2Cursor, 242
setFillColor, 241

setIntegralFillMode, 237
setMargins, 265
setMaxContourObjects, 233
setPenColor, 241
setPenStyle, 241
setPenWidth, 240
setPolylineMode, 236
setView, 232
showAllContours, 234
showContour, 233
ShowCursor, 228
ViewChanged, 245
ViewMoveZoomMode, 230
SIEsTouchButton, (Language-dependent texts)
BackColor, 259
BackColorChecked, 259
BackColorDisabled, 260
BackColorPressed, 260
BackgroundPicture, 262
BackgroundPictureAlignment, 263
BackgroundPictureAlignmentString, 264
BackgroundPictureKeepAspectRatio, 265
ButtonStyle, 250
Checkable, 252
Checked, 252
clicked, 267
clickedDisabled, 268
Enabled, 251
Flat, 250
Overview of functions, 265
Overview of signals, 266
Picture, 253
PictureAlignment, 254
PictureAlignmentString, 255
PictureKeepAspectRatio, 256
PicturePressed, 254
ScaleBackgroundPicture, 264
ScalePicture, 255
ShowFocusRect, 253
Text, 256
TextAlignedToPicture, 259
TextAlignment, 257
TextAlignmentString, 258
TextColor, 261
TextColorChecked, 261
TextColorDisabled, 262
TextColorPressed, 262
TextPressed, 256
Signal color"; "progress bar, 95
Signal values"; "progress bar, 92
SIEsTouchButton
Reading and writing properties, 248
slhlp.xml, 77
SIX3dViewerWidget, 333
Softkey
Assign properties, 62
Properties, 64
Start softkey, 31, 33
Strings, 106
Sub-dialog, 165
Subprogram, 138
Block identifier, 141
Call, 140
cancel, 175
Variable, 141
System colors, 309
System variable, 83, 95

## T

Table
Defining columns, 204
Definition, 202
Programming, 203
Text, 92
Toggle field, 87, 92, 101
Toggle symbol, 93
Tooltip, 92, 94
Touch operation, 247
Trigonometric functions, 122

## U

Unit text, 92
Update rate, 93
User variable, 95

## V

V_SEPARATOR (define vertical dividing line), 194
Variable
calculating, 82
Change property, 96
Check, 142
CURPOS, 108
CURVER, 109
End, 152
ENTRY, 110
ERR, 111
FILE_ERR, 112
FOC, 114
Parameter, 92
S_ALEVEL, 115

S_CHAN, 116
S_CONTROL, 117
S_LANG, 118
S_NCCODEREADONLY, 119
S_RESX, 120
S_RESY, 120
Variable status, 81
Variable type, 92
INTEGER, 98
VARIANT, 99
Variable value, 81

## W

Write mode, 95

SIEMENS

System settings 1

Configuring the system 2

Service and diagnostics 3

Appendix A

# SINUMERIK

# SINUMERIK 840D sl
# NCU operating system

Commissioning Manual

Valid for:

CNC software for 840D sl/ 840DE sl Version 4.8 SP2

12/2017
6FC5397-1DP40-6BA2

## 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

6FC5397-1DP40-6BA2
℗ 08/2018 Subject to change

Copyright © Siemens AG 2006 - 2017.
All rights reserved

# Table of contents


**1 System settings....5**

1.1 Delivery condition of the system....5

1.2 Meaning of switch positions....8

1.3 System booting....9

1.4 Displays during system booting....10

**2 Configuring the system....13**

2.1 Ethernet interfaces of the NCU....13

2.2 How to determine the IP address of the NCU....14

2.3 Configuration file "basesys.ini"....15
2.3.1 Section [ExternalInterface]....16
2.3.2 Section [InternalInterface]....18
2.3.3 Section [IBNInterface]....21
2.3.4 Section [SNMP]....22
2.3.5 Section [DCP]....24
2.3.6 Section [LLDP]....24
2.3.7 Section [LinuxBase]....25
2.3.8 Example: Configuration file 'basesys.ini'....30

2.4 Service Commands....38
2.4.1 Using service commands....38
2.4.2 Syntax for the Actions....39
2.4.3 Generic elements....40

2.5 Description of the actions....42
2.5.1 Help....42
2.5.2 Check-cf....43
2.5.3 Clear....43
2.5.4 Closeport....44
2.5.5 Connect....44
2.5.6 Disable....47
2.5.7 Disconnect....48
2.5.8 Distribute....48
2.5.9 Enable....49
2.5.10 ntpdiag (diagnostics)....50
2.5.11 Openport....51
2.5.12 Portstatus....52
2.5.13 Restart....52
2.5.14 Restore....53
2.5.15 Save....54
2.5.16 save-nvram (data backup)....55
2.5.17 set-svc-route (gateway)....56
2.5.18 Show....56
2.5.19 Start, Stop....60

2.5.20 usb (connect device)....61
2.5.21 vncpwd (password)....62

**3 Service and diagnostics....63**

3.1 Creating a service system....63
3.1.1 How do you create a service system for the NCU?....63
3.1.2 This is how you operate the service system....65
3.1.3 Diagnostics functions....66
3.1.4 This is how you connect to a network drive....67

3.2 Backup license....69
3.2.1 This is how you backup the license key....70

3.3 Backup data via the service menu....72
3.3.1 This is how you backup user data....72
3.3.2 This is how you restore user data....75
3.3.3 This is how you create a complete backup of the CompactFlash Card....76
3.3.4 This is how you install a complete system backup....78

3.4 Backing up data via the service shell....80
3.4.1 This is how you save data on a service system....80
3.4.2 This is how you restore data from the service system....81
3.4.3 This is how you save data on a network drive....81
3.4.4 This is how you restore data from the network drive....82

3.5 Updating the software....84
3.5.1 This is how you update the firmware....84
3.5.2 This is how you update the software....86

3.6 System diagnostics....89
3.6.1 Performing system diagnostics....89
3.6.2 This is how you change the system settings....91
3.6.3 This is how you check the CompactFlash Card....92
3.6.4 Calling the VNC Viewer....94
3.6.5 WinSCP and PuTTY....96

**A Appendix....97**

A.1 Abbreviations....97

**Glossary....99**

**Index....101**

# System settings

# 1

## 1.1 Delivery condition of the system

### Scope of validity

This document is valid for all systems that are supplied with Linux as operating system on the CompactFlash Card, such as all NCUs of the type NCU 7x0.3B PN.

### Directories on the CompactFlash Card

A Linux partition as well as a FAT partition are available on the CompactFlash Card. The memory space of the CompactFlash Card is occupied mainly by the Linux-EXT3 partition containing the system software and the user data. A 2 MB large FAT partition still exists but only for internal administration purposes.

When the system is in operation, the following directories (this is a selection and not a complete list) are available in the EXT3 partition on the CompactFlash Card:

| Directory | Use |
|---|---|
| /siemens | Reserved for Siemens system software |
| /addon | Reserved for Siemens add-on software |
| /oem | Additional software and configurations of the machine manufacturer |
| /user | • Storage of user data<br>• Configuring the HMI application<br>• Data that is created when commissioning the system using the HMI application. |
| /system | Linux operating system |
| /user/system/etc | File basesys.ini (modifications possible) |
| /user/common/tcu | TCU configuration files |
| /var/log/messages | System log file (same as event.log under Windows) |

**Note**

Files in the directory under /user always have priority over files with the same name in the directory /oem → /addon → /siemens.

## Partitioning of the memory

The memory of the CompactFlash Card is purely logically partitioned, i.e. the user can in principle assign the memory up to the end of the physical limit. An alarm under LINUX is triggered if less than 4 MB free memory is available.

| Directory | 1 GB card | 8 GB card |
|---|---|---|
| /oem<br>/addon | 100 MB | 1 GB |
| /user | 256 MB | 2 GB |

## Preset users

The following users have already been set up:

| Users | Access level | Target group |
|---|---|---|
| Operator | Key switch position 0 | Operators, users |
| Operator1 | Key switch position 1 | Operators, users |
| Operator2 | Key switch position 2 | Operators, users |
| Operator3 | Key switch position 3 | Operators, users |
| user | Password:CUSTOMER | Operators, users |
| service | Password: EVENING | Service personnel |
| manufact | Password:SUNRISE | Machine manufacturer |

**NOTICE**

**Changing the password**

You can only change the passwords of Linux users in the basic NCU system - not for the booted service system.

To increase system security, you should change the passwords in the Linux system for users "manufact", "service" and "user" analogous to the passwords for the NC access levels.

You can access the basic NCU system e.g. via the WinSCP/Putty (Page 96) program, via a service console or on a TCU.

Call the service console on the TCU with the menu back key and key <MENU SELECT>:



### Rules relating to the password structure

The following rules must be observed when assigning a new password:

- Permissible (and specified) characters include lowercase letters, uppercase letters, numbers and special characters.
- The password must have a minimum length of 6 characters. For each missing character category (lowercase letters, uppercase letters, numbers and special characters), the minimum length is extended by two characters.
  Example: As a consequence, a password comprising only uppercase letters must be 12 characters long.
- It is not permissible that the password contains too many identical characters
- It is not permissible that the password has any similarity to the user name, the computer name and the old password (e.g. write the password backwards, interchange uppercase/ lowercase letters).

If this rule is not complied with, then "too weak" message is output – and the old password remains unchanged.

### Changing the password

You can only change the user password with which you used to login; this means that if you wish to change the passwords of all default users, then you must login with the particular login one after the other.

```
$ passwd
Changing password for service
Old password:
New password:
Bad password: too weak
passwd: password for service is unchanged
$ passwd
Changing password for service
Old password:
New password:
Retype password:
```

Procedure:

1. Log into the basic NCU system (Linux) with user name and password.
2. Enter the "passwd" command.
3. Follow the instructions to enter the old, previous password.
4. Enter a new password and repeat this to confirm the password.
5. The new password is valid once all of the entries are correct. Any deviation means that the dialog is exited, and the old password remains unchanged and valid.

# 1.2 Meaning of switch positions

## Overview

The NCU has two twist buttons in the lower section of the front panel.

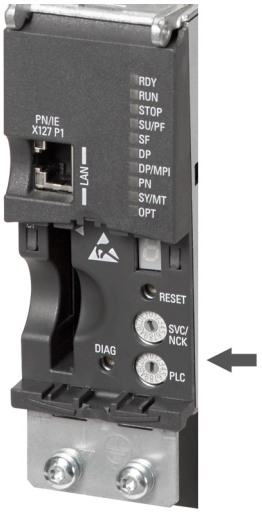



Figure 1-1 Diagnostic displays of the NCU

## Rotary switch SVC/NCK

The switch positions have the following significance:

| Switch position | Operating mode of the NCK |
|---|---|
| 0 | Normal boot-up NCK |
| 1 | NCK boot-up with default values (= memory reset) |
| 2 | The NCK (and PLC) starts up with the data that was saved at the last shutdown. |
| 7 | Debug mode (NCK is not started.) |
| 8 | IP address of the NCU is displayed on the seven-segment display. |
| All others | Not relevant |

## PLC twist button

The switch positions have the same significance as for a SIMATIC S7-CPU:

| Switch position | Operating mode of the PLC |
|---|---|
| 0 | RUN |
| 1 | RUN (protected mode) |
| 2 | STOP |
| 3 | Memory reset (MRES) |
| All others | Not relevant |

# 1.3 System booting

## Sequence

To ensure unproblematic booting of the NCU, the CompactFlash Card must be inserted.

When the NCU is booting up, visual information on the current operating system is provided using the following displays:

- The RDY-LED flashes slowly orange when the CF card is accessed.
- During booting, the 7-segment display outputs different codes that indicate, for example, when the BIOS is started, when the CompactFlash Card is accessed, etc.

When the booting has been completed successfully, the following is displayed:

- The PLC LED lights up green.
- The 7-segment display shows "6." with a flashing dot.
- The RDY-LED lights up green, all other LEDs remain dark.

## Performing a reset operation

The reset button is located behind the blanking plate of the NCU.

A reset operation resets the entire system and requires a system restart. This is comparable to a "Power On reset" - except that the 24 V power supply does not have to be switched off.

## Booting for servicing purposes

For service or diagnosis purposes, the NCU can be booted from a service system, the Emergency Boot System.

## See also

How do you create a service system for the NCU? (Page 63)

# 1.4 Displays during system booting

## Behavior of the LEDs when booting

Of the LEDs on the front panel of the NCU, only the RDY LED and its status is important when booting the NCU.

| RDY LED | Additional LEDs | State/phase | 7-segment display |
|---|---|---|---|
| Orange | Orange | BIOS boot | Post codes<br>See table: Load BIOS |
| Slowly flashing orange (0.5 Hz) | Set from the PLC or option module | Booting up the kernel<br>Phase 2: Initialization of the driver | 1 |
| Slowly flashing orange/green (0.5 Hz) | Set from the PLC or option module | Booting the basic system<br>Phase 3: Initialization of the basic system | See table: Load the basic system |
| OFF | Set from the PLC or option module | NRK/NCK outputs | See table: NRK/NCK outputs |
| green/red flashing 0.5 Hz) | Set from the PLC or option module | Basic system shutdown | OFF |
| red | Set from the PLC or option module | Kernel shutdown | OFF |
| red (5 Hz) | Set from the PLC or option module | SINAMICS fault | --- |

SINAMICS faults and alarms, also see: SINAMICS S120/S150 List Manual (LH1)

## Load BIOS

| 7-segment display | Meaning |
|---|---|
| 00 | Before loading the Master Boot Record (MBR) from the CompactFlash Card |
| 01 | Before starting the Master Boot Record (MBR) codes |
| 02 | Master Boot Record (MBR) code started |
| 03 | Second stage of the boot loader started, before reading the configuration file |
| 04 | Before loading the kernel |
| 05 | Before starting the kernel |

## Booting the basic system

See diagram: Figure 1-1 Diagnostic displays of the NCU (Page 8)

| 7-segment display | Meaning |
|---|---|
| 1 (begin) | Basis initializations |
| 2 (prepfs) | Cleaning up and preparing the file system |
| 3 (hostname, loopback) | The name is set and the loopback interface configured. |
| 4 (syslog) | Starting the system logging<br>Starting the CFS class 1, provide the network interfaces. |

| 7-segment display | Meaning |
|---|---|
| 5 (network) | Initializing the network interfaces and the firewall |
| 6 (NCU:rtai, TCU:tcu-config) | Starting the time server (ntpd) |
| 7 (lsh) | Starting the SSH server (generate a host key if one is still not available). |
| 8 (NCU:ftpd, TCU:mtouch) | Starting the FTP server |
| 9 (NCU:tcuservices, TCU:usbexport) | Starting the TCU services (TFTP, VNC) |
| U r | Update required:<br>The CompactFlash card of the NCU contains a software version that is not executable on type 3B modules and cannot be updated by an update.<br>**Remedy:**<br>• Check whether this CompactFlash card runs up in another NCU.<br>• Check whether the CompactFlash card is defective with the aid of a service system: sc check-cf (Page 43)<br>The code "U r" can also be output in the following cases:<br>• No CompactFlash card is inserted.<br>• The CompactFlash card cannot be read.<br>• The partition table of the CompactFlash card is defective. |

## NRK/NCK outputs

See diagram: Figure 1-1 Diagnostic displays of the NCU (Page 8)

| 7-segment display | Meaning |
|---|---|
| 1 | Not applicable. |
| 2 | Not applicable. |
| 3 | Debugger was initialized. |
| 4 | NRK operating system was successfully activated. |
| 5 | NRK operating system has booted, init task is being processed (→ the application boots). |
| 6 | Initialization was successfully performed; the control is in cyclic operation. |
| 6. | with flashing point: The control is in cyclic operation and the cycles are active. |
| F | Internal error: Can only be read via additional diagnostics. |
| 1 or 2 | CompactFlash Card and SRAM data do not match. To resolve this problem, a general reset is required by booting with switch position 1 or 2. |
| Lxx=yyy | Internal error when starting NCK, "xx" is the line number in the mcsystem.ini, "yyy" is a unique error code for the code position. |

## System error

See diagram: Figure 1-1 Diagnostic displays of the NCU (Page 8)

| RDY LED | Additional LEDs | Meaning | 7-segment display | Meaning |
|---|---|---|---|---|
| Rapidly flashing red (2Hz) | Set from the PLC or option module | **Permanent error:**<br>The system is stopped. | C | "crash": The operating system crashes (this approximately corresponds to a Blue Screen for Windows); details can be found in the system log file. |
| | | | P | "partition": Error when repartitioning |
| Rapidly flashing red/orange (2 Hz) | Set from the PLC or option module | **Temporary error:**<br>The system may continue to run with restricted functions. | E | "error": A read or write error has occurred on the CompactFlash Card (an additional point is displayed for a write error). It is possible that the CompactFlash Card is defective and should be replaced. |
| | | | F | "full": The CompactFlash Card is too full, so it is probable that not all of the services will be able to be successfully started.<br>When booting, this error is displayed for 1 minute; the system then continues to boot, however, it can be assumed that problems will occur. |

# Configuring the system 2

## 2.1 Ethernet interfaces of the NCU

### Supplementary conditions

For the operation of an NCU:

- No more than one NCU may be operated as a DHCP server on the system network.
- An external mouse is recommended when using the System Network Center.

### Configuration of the interfaces

The following connections can be established via the Ethernet interfaces:

| Interface | Labeling | Inter-nal name | Terminal settings |
|---|---|---|---|
| Ethernet IE1/OP | X120 | (Eth 2) | Connection to the system network with preset IP address 192.168.214.1 with subnet screen form 255.255.255.0 and active DHCP server for SINUMERIK |
| Ethernet IE2/NET | X130 | (Eth 1) | Connection to company network as standard DHCP client |
| Ethernet | X127 | (Ibn 0) | Service terminal with fixed IP address 192.168.215.1 and fixed subnet screen form 255.255.255.224 with active DHCP server |

### Network interface

The network interface is an interface that enables network communication. These are the Ethernet interfaces on the NCU.

### VNC (Virtual Network Computing)

Virtual Network Computing is a software that displays the screen contents of a remote computer, with a running VNC server, on a local computer, with a running VNC viewer, and in return sends keyboard and mouse movements of the local computer to the remote computer.

## 2.2 How to determine the IP address of the NCU

### NCU in the system network (X120)

On the X120, the NCU is pre-selected as required: No settings are necessary.

### NCU in the company network (X130)

On the X130, the NCU is set to the address reference via DHCP. If the company network has a DHCP server no further settings are required.

### Example: Determining the IP address

There are several possibilities for determining the address that the NCU obtained in the company network:

- If the NCU boots in switch position 8, it shows the IP address on the X130 on the 7-segment display.
- Once the NCU has booted successfully, open a service shell on the TCU and execute the following command (Page 56) to obtain the desired information:
  `SC SHOW IP`

If the company network has no available DHCP server or if this cannot be used, you are given the option to set a fixed address for the NCU on the X130 (in accordance with the addresses already used in the company network).

The following command displays the IP address on the X130:

```
SC SHOW IP -X130
```

**Note**

The IP addresses can also be set directly from the SINUMERIK Operate user interface in the "Commissioning" → "Network" operating area.

# 2.3 Configuration file "basesys.ini"

## Storage path

For a Linux basic system, a template of the file is provided on the system CompactFlash card under the following path:
`card/siemens/system/etc/template-basesys.ini`

Copy this template with the name basesys.ini into one of the following directories:
`oem/system/etc/basesys.ini`
`user/system/etc/basesys.ini`

## Use

Settings are made in the basesys.ini file that specify the behavior of the system in the network.

To edit the basesys.ini file:

- Comments are introduced by ';' or '#' at the beginning of the line, and span the width of the line. Empty lines are also handled as comments.
- The sections that start with a "[NAME]" line are ignored by the basic system itself, but are sometimes used by the HMI.
- Variable definitions are written in the form "NAME=VALUE". Blanks are permitted around the '=' character. The value may also be enclosed in double quotation marks, but this is optional.

**Note**

**Suitable editors for Linux**

In most Linux system files, lines may only be ended with LF, and not with CRLF as in Windows. Please take note of this when selecting an editor. The HMI application editor under "Commissioning" is suitable.

In the Linux operating system, the UNIX editor vi is available.

Please take note that the Linux operating system is case-sensitive.

## Reference

To make settings in the basesys.ini file, dialogs are provided in the "Commissioning" operating area in SINUMERIK Operate in order to display or modify the values of the individual variables: Commissioning Manual "SINUMERIK Operate" (IM9)

## 2.3.1 Section [ExternalInterface]

### Description

In this section, the parameters and settings for the external Ethernet interface are defined. For an NCU7x0, this is X130 (Eth1). If the external interface is in DHCP mode, i.e. the value of ExternalIP is empty or not defined, then all parameters listed here, with the exception of "DHCPClientID", are accepted by the DHCP server as long as the server provides a value for the parameter.

### DisableCompanyNet

If this variable is set to 1, the X130 interface to the company network is not initialized and is thus not available.

Value: Empty, 0 or 1
Default: Empty

### ExternalIP

If ExternalIP is not empty, the fixed IP address specified there is used in the company network interface. The ExternalNetMask then also needs to be set, as well as the gateway, name servers, host name, and domain if required.

If ExternalIP is not set or is empty, a DHCP client is started on the interface.

Value: IP address
Default: Empty

### ExternalNetMask

ExternalNetMask must be set together with ExternalIP, in order to define the size of the network.

Value: Subnet mask
Default: Empty

### Gateway

If the value of gateway is not empty, the host specified there is used as the default gateway, which means that all IP packets that cannot be directly assigned are sent there for routing.

If no gateway is specified, only the networks that are directly connected are accessible.

Value: IP address
Default: Empty

## Nameservers

If DNS name servers are specified here, then these must be used to resolve symbolic host names, i.e. at most of the points where an IP address is expected, a computer name can also be used instead.

The name server setting is also sent to your DHCP client (TCU, PG) via the DHCP server of the NCU, so that this can also work with symbolic names.

Value: List of IP addresses (separated by blanks)
Default: Empty

## Timeservers

Here you can specify a list of NTP servers (Network Time Protocol) that can be used by the NTPD on the NCU for time synchronization. The status of NTPD is output in the system log file /var/log/messages

As an alternative to the system log file, you can use "ntpq -p localhost" to find a time server marked '*'. The time is synchronized with this server.

Value: List of IP addresses (separated by blanks)
Default: Empty

**Note**

**Setting the time zone and the time server**

The IP addresses can also be set directly from the SINUMERIK Operate user interface in the "Commissioning" → "Date time >" softkey operating area.

## Hostname

Here you can define a name for the local host. This manually assigned name takes priority over all other names.

The host name is determined in the following order:

- Host name from the basesys.ini file, if set.
- A name received from the DHCP server (if DHCP client mode and a name have been supplied), the result of a reverse DNS lookup, i.e. which name belongs to the IP received (if name servers are defined).
- Default name ("NONAME_...")

Because the host name is also used as DNS name it must satisfy the corresponding requirements of the RFC:

- ASCII letters (a-z, A-Z), numbers (0-9) and '-'
- max. 63 characters

Value: Name
Default: Empty

### Domain

These variables can be used to configure the DNS domain. The effect is that names in this domain can be resolved without qualification (e.g. if the domain is "test.local", you can also specify the name "computer1.test.local" as "computer1").

Value: Name
Default: Empty

### DHCPClientID

With these variables, you can influence which ClientID the DHCP client presents to its server. This ID can be used by the server to assign certain parameters to the client, for example a static IP address.

Usually, the MAC address of the Ethernet interface is used for this purpose. This is also the default setting. Alternatively, you can also use the host name ("@NAME"), which must then be defined in the basesys.ini, in order to be recognized before the DHCP request. It is also possible to use an arbitrary string for ClientID.

Value: @MAC, @CFID, @NAME or any string
Default: @MAC

## 2.3.2 Section [InternalInterface]

### Description

In the section [InternalInterface] the data is specified for synchronization of the DHCP server in the system network.

### DisableSysNet

If this variable is set to 1, the X120 interface to the system network is not initialized and is thus not available.

Value: Empty, 0 or 1
Default: Empty

### InternalIP

Use this variable to set the fixed IP address in the system network. InternalIP should always be used together with InternalNetMask.

Value: IP address
Default: 192.168.214.1

### InternalNetMask

Specifies the subnet mask for InternalIP and should always be used together with it.

Value: Subnet mask
Default: 255.255.255.0

### InternalIP_Alias

Use this variable to define an additional alias IP address for the interface of the system network (X120); this is practical in certain application cases.

Value: IP address
Default: Empty

InternalIP_Alias should always be used together with InternalNetMask_Alias.

### InternalNetMask_Alias

This variable specifies the subnet mask for InternalIP_Alias and should always be used together with it.

Value: Subnet mask
Default: Empty

### EnableDHCPD_SysNet

If this variable is set to zero, then the start of the DHCP server in the system network X120 is prevented.

Value: Empty, 0 or 1
Default: Empty

### SyncModeDHCPD_SysNet

If this variable is not set to "OFF", the DHCP servers synchronize themselves in the system network (X120) so that only one of them actively assigns addresses. This makes it possible to operate multiple NCUs or PCUs concurrently without having to adjust the network settings.

Value: ON_MASTER, ON_HIGH (= ON), ON_LOW, or OFF
Default: ON_HIGH

The DHCP servers that are not active go into "standby" mode; in this mode they regularly get the current address data and TCU data from the active server so that if the active server fails a standby server can take over the active role without data loss.

- With the **ON_MASTER** setting instead of **ON_HIGH** you can influence the synchronization in such a manner that the server with the "Master" setting is always the active server (for this it must be active in the network). This means that in the normal situation deterministically the same control is always the DHCP server and you can find the actual address data (/var/etc/udhcp-eth0.leases) and the TCU data (/user/common/tcu/...) there. ON_MASTER should only be set for a single DHCP server in the system network; there should not be multiple masters.
- The **ON_LOW** setting assigns the DHCP server a lower priority than normal. It then becomes the active server if no server with ON_HIGH or ON_MASTER is found in the network.

**Note**

**Compatibility with earlier versions**

The priority level "ON_LOW" is used automatically if the DHCP server is switched off. Although such an NCU does have a (standby) DHCP server and keeps a backup of the lease data, normally, however, it does not become an active server.

### InternalDynRangeStart

With the two variables InternalDynRangeStart and InternalDynRangeEnd, you can explicitly specify the range of IP addresses assigned by the DHCP server. The number range of the presetting should normally suffice.

Value: IP address
Default: First address in the system network + 10 or +2 (if ≤ 16 addresses)

### InternalDynRangeEnd

Value: IP address
Default: Last address in the system network - 15 or -1 (if ≤ 16 addresses)

### DHCPDNoMasterWait

If a synchronizing DHCP server has once "detected" a master server in the past, then it waits for synchronization until the time specified here elapses before it becomes the active server.

This additional pause makes it possible for the intended DHCP master to become the active DHCP server without displacement even if it is shortly switched on thereafter, or takes longer to boot than other controls.

Value: Time in seconds
Default: 120

### InternalDNSDomain

This is used to specify the Top Level Domain (TLD) name that is used in the system network. The DNS server of the NCU assigns names to the devices in the system network in this zone. Name requests for all other zones are forwarded to an external name server in the company network.

As presetting the "local" recommended by the RFC1035 is used for local networks, to avoid conflicts with globally defined domain names.

Recommendation: The default setting should be maintained.

Value: Domain name (letters, digits, '-', '_', max. 63 characters)
Default: local

### DisableNATRouting

The default setting permits that a machine can access a company network from the system network via the control using the NAT routing (= "Network Address Translation"). In this case, the NCU converts the source address of the packets to its own address, and handles the response packets in the inverse way.

If this variable is set to zero, the NAT routing of the NCU from the internal interfaces eth0 =X120 and ibn0 =X127 to the external network on eth1 =X130 is prevented. In order that the NAT routing is not simultaneously shut down for both internal networks, using the values "X120" or "X127", the system network or the commissioning network is selectively shut down.

Value: empty, 0, 1, X120 or X127
Default: Empty

## 2.3.3 Section [IBNInterface]

### Description

The settings in this section affect interface X127.

### EnableDHCP_IBNNet

If this variable is set to 1 then this prevents the start of the DHCP server at the commissioning interface X127.

Value: Empty, 0 or 1
Default: Empty

### DisableIBNNet

If this variable is set to 0, then the commissioning interface X127 is not initialized and is thus not available.

Value: Empty, 0 or 1
Default: Empty

### EnableSysNetToIBNForwarding

If this variable is set to 1, it is possible to transfer packets from the system network (X120) to X127. A firewall usually prevents this.

It is important to remember that there is no NAT involved, and the sender is responsible for ensuring (with a suitable routing entry, for example) that packets sent to the PG or service PC on X127 with the address 192.168.215.x only reach the NCU to which the device is connected.

Value: Empty, 0 or 1
Default: Empty

| NOTICE |
| --- |
| **Security risk**<br>Note that opening the firewall can represent a security risk. Permit feed-through only for the period for which it is actually required. Note that every accessible service can have security gaps. |

### DisableIBNForwarding

If this variable is set to 1, the NAT transfer of packets from the service connection X127 to the system network (X120) is deactivated.

Value: Empty, 0 or 1
Default: Empty

## 2.3.4 Section [SNMP]

### Description

Character strings delivered via SNMP (Simple Network Management Protocol) are entered here. This is a way of giving out information.

### SNMPLocation

This string is used for the standard OID SNMPv2-MIB::sysLocation. It is possible to specify a location here, which can be subsequently called with an SNMP client.

Value: Any character string
Default: Empty

### SNMPContact

This string is used for the standard OID SNMPv2-MIB::sysContact. It is possible to specify a contact address here, which can be subsequently called with an SNMP client.

Value: Any character string
Default: Empty

### SNMPAutLocation

This string is used for the Siemens-specific OID automationSystemIdent.automationLocationTag. It is possible to specify an additional location here, which can be subsequently called with an SNMP client. This is similar to SNMPLocation in terms of the MIB description, although it is intended for entries relating to automation.

Value: Any character string
Default: Empty

### SNMPFunction

This string is used for the Siemens-specific OID automationSystemIdent.automationFunctionTag. It is possible to specify a function designation here, which can be subsequently called with an SNMP client.

Value: Any character string
Default: Empty

### SNMPStation

This string is used for the SINUMERIK-specific OID mcSinumerikMIB.mcSinumerikMiscStation. The value is the name of a station to which the device belongs. Devices with the same station name are therefore identified as being contiguous.

Value: Any character string
Default: Empty

## 2.3.5 Section [DCP]

### Description

The parameters in this section define the properties of the DCP protocol (Discovery and Basic Configuration Protocol), e.g. that are used in the menu item "Accessible nodes" of STEP 7.

### InternalDcpEnabled

This can be used to switch-on and switch-off the DCP in the system network (X120); for the NCU, the value is also used for X127.

Value: 0 or 1
Default: 1

### ExternalDcpEnabled

This can be used to switch-on and switch-off the DCP in the company network (X130).

Value: 0 or 1
Default: 0

## 2.3.6 Section [LLDP]

### Description

The parameters in this section define the properties of the LLDP protocol (Link Layer Discovery Protocol), which is used by several applications for network diagnostics.

### InternalLldpEnabled

This can be used to switch-on and switch-off the LLDP in the system network (X120); for the NCU, the value is also used for X127.

Value: 0 or 1
Default: 1

### ExternalLldpEnabled

This can be used to switch-on and switch-off the LLDP in the company network (X130).

Value: 0 or 1
Default: 0

### InternalLldpTLVsTxEnabled

Using this parameter, additional information in the LLDP packets are enabled at X120/X127, which are normally not included.

Value: Numerical value from 0 ... 15
Default: 0

The value is a bit field, i.e. the total value is obtained as a sum (total) of the specified numbers, if the corresponding information is to be sent:

1: port description
2: system name
4: system description
8: capabilities

### ExternalLldpTLVsTxEnabled

Using this parameter, additional information in the LLDP packets is enabled on the X130, which are normally not included.

Value: Numerical value from 0 ... 15
Default: 0

The value corresponds to parameter InternalLldpTLVsTxEnabled.

## 2.3.7 Section [LinuxBase]

### Description

Additional setting possibilities of the Linux system are combined in this section.

**Note**
**Setting the time zone and the time server**

The IP addresses can also be set directly from the SINUMERIK Operate user interface in the "Commissioning" → "Date time >" softkey operating area.

## Synchronizing the time

You must distinguish between the following cases when synchronizing the time:

- If there is an external NTP server to act as a time server (entered in the basesys.ini file or via DHCP), then the PLC clock is synchronized via the Linux clock.
- If there is no external time server, the PLC clock will be the master for the Linux clock: With "memory reset" the time of the PLC clock remains intact and is not reset. If the PLC is switched on in switch position 3 "MRES" then the time will be reset. In this case the valid time is taken over from the Linux system and does not need to be reset.

You can obtain information about the NTP server using the following command: sc ntpdiag (Page 50)

| NOTICE |
| --- |
| **Restriction on time synchronization**<br>Do not set the time synchronization on two different systems; this means:<br>Select the NTP server either in the PLC project via PROFINET IO (X150) or in the "basesys.ini" on the NCU in section [LinuxBase], but not in both simultaneously, as an additional time zone can be specified on the NCU - however, not in the PLC project. |

## Timezone

The time zone set here is used by the system to convert UTC time (Universal Time Coordinated) to local time. The time zone is sent to all applications via the environment variable "TZ", and is taken into account by the libc function localtime().

The time zone influences all time entries in the basic system, particularly for the command "`date`", for "`ls -l`", and in the system log file /var/log/messages.

Value: Time zone description (see syntax and examples)
Default: UTC

**Examples:**

If the control is to synchronize with one of the listed time zones, the examples in the table can simply be copied.

**Zones with daylight saving time/standard time:**

| | |
| --- | --- |
| Europe: | WET0WEST,M3.5.0,M10.5.0/3 |
| | CET-1CEST,M3.5.0,M10.5.0/3 |
| | EET-2EEST,M3.5.0,M10.5.0/3 |
| USA: | EST5EDT,M4.1.0,M10.5.0 |
| | CST6CDT,M4.1.0,M10.5.0 |
| | MST7MDT,M4.1.0,M10.5.0 |
| | PST8PDT,M4.1.0,M10.5.0 |

**Zones without daylight saving time/standard time:**

China: CST-8
Japan: JST-9

### Syntax for the time zone description

In order to synchronize with a different time zone, the following syntax applies:

tz offset dst,Mmwd,Mmwd

| Syntax (obligatory): | |
|---|---|
| tz offset | Time zone and difference from UTC:<br>West of the zero meridian, the difference must be added; east of the zero meridian, the difference must be subtracted. |
| Syntax (optional, only if there is daylight saving time): | |
| dst,Mmwd,Mmwd | Daylight saving time zone with beginning and end |
| m (1 ≤ m ≤ 12) | m: Month |
| w (1 ≤ w ≤ 5) | w: Week of the month, i.e. w = 5 is the last week of the month |
| d (0 ≤ d ≤ 6) | d: Day of the week (from 0 = Sunday to 6 = Saturday) |

## PLCTimeType

This variable specifies the type of time that the PLC clock has been set to:

- `local-nodst` stands for local time **without** daylight saving time, even if the time zone defines daylight saving time. In summer, all clocks are 1 hour behind the normal time, but as a consequence there are no time steps. If the PLC clock is inadvertently set to daylight saving time, then the Linux clock advances by 1 hour.
  This behavior was used until this variable was introduced, and for compatibility reasons, it still remains the default setting.
- `local` b means local time **with** changeover to daylight saving/standard time if in the corresponding time zone there is daylight saving. If Linux takes its time from an external server, then with this setting, the PLC clock is changed over twice a year.

  **NOTICE**

  **Time of day alarms of the PLC missing or are duplicated**

  If Linux takes its time from the PLC, then this must either be manually or automatically changed over externally between daylight saving time and standard time. If this is not done, then in summer, the Linux clock is back 1 hour.

- `utc` means that the PLC clock is set to universal time. This makes it easy to align the time as also the Linux clock is internally set to UTC. For this setting, there are no time changes as UTC never has daylight saving time. However, neither SINUMERIK Operate nor the engineering tools of the PLC know this mode or support converting UTC into local time, so that in most cases, a time differing from the normal time is displayed.

Value: local-nodst, local or utc
Default: local-nodst

### LogFileLimit

The size of the system log file `/var/log/messages` can be restricted with this variable. By entering "Zero", the log file has no size limitations.

The specified limit is no hard limit for reasons of efficiency but rather a sort of target value. If the log file reaches the target value times factor 1.5, then it is reduced to around 75% of the target value. A check is carried out every two minutes maximum to see if the file has become too large.

Value: Size in KB
Default: 100 KB

### FirewallOpenPorts

Here you can specify a list of the ports which is to be enabled in the firewall of the external network interface. The protocol is "TCP" or "UDP", the port is the (numerical) port number of the service to be enabled. Several entries are separated by blanks.

Value: List of protocol/port pairs
Default: Empty

| NOTICE |
| --- |
| **Security risk** |
| Note that opening the ports in the firewall can represent a security risk. Only enable the ports that you actually require. |
| Note that every accessible service can have security gaps. |

### DisableSubsystems

Using this variable, you can suppress mounting and starting of certain subsystems (CFS systems). Several subsystem names can also be specified - separated by blanks.

Value: List of CFS names (separated by blanks)
Default: Empty

The names can either be simple CFS basic names (e.g. "nck"), whereby all CFS systems with this name are ignored. Alternatively you can specify an absolute path (e.g. `/siemens/sinumerik/nck`) referring specifically to this CFS system.

It is also possible to set "DisableSubsystems=all" whereby not a single CFS system can be mounted or started any more.

### DefaultCIFSSecurity

The Linux command sc connect, has for CIFS network drives, an option "-sec=TYPE"; with this option, the security type can be set when establishing a connection to the server. If, when calling sc connect (Page 44) this option is not specified, then as a substitute, the value of variable DefaultCIFSSecurity is used in basesys.ini:

| | |
|---|---|
| none | No authentication, connection only as guest – if the server permits this. |
| ntlm | NTLMv1 password hashing (as a result of compatibility) |
| ntlmi | NTLMv1 with forced package signing (due to compatibility) |
| ntlmv2 | NTLMv2 password hashing |
| ntlmv2i | NTLMv2 with forced package signing |
| ntlmssp | NTLMv2 via NTLMSSP packaging |
| ntlmsspi | NTLMSSP with forced package signing |

(krb5 and krb5i - as a result of the missing Kerberos infrastructure, are not supported on the NCU.)

| | |
|---|---|
| Value: | none, ntlm, ntlmi, ntlmv2, ntlmv2i, ntlmssp, ntlmsspi |
| Default: | ntlmssp |

### DisableNTPTimeSync

If this variable is set to 1, then the NTPD server (Network Time Protocol Daemon) for the time synchronization is not started and the time of day is not synchronized.

| | |
|---|---|
| Value: | 0 or 1 |
| Default: | 0 |

### DisablePLCTimeSync

If this variable is set to 1, synchronization of the system time with the PLC time (in both directions) will not be executed.

| | |
|---|---|
| Value: | 0 or 1 |
| Default: | 0 |

### EnableCoreDumps

If this variable is set to 1, a log file for processes is created and stored under `/var/tmp`. The log file contains a memory dump of the process that has crashed or that has terminated.

| | |
|---|---|
| Value: | Empty, 0 or 1 |
| Default: | 0 |

### CheckTCUforUpdatePackage

If this variable is set to 1, when booting, the USB interface at the front of the TCU is checked to identify any storage medium. If, when checking the update packages, type *.usz files created with Create MyConfig are found, then the installation of these update packages is started.

With the default setting zero, this check is skipped in order to save time.

Value: 0 or 1
Default: 0

### EnableUsbShares

If this variable is set to 1, then USB memories are integrated as jointly used network drive.

If this variable is set to 0, then USB memories are integrated as individual USB data storage device.

Value: 0 or 1
Default: 1

## 2.3.8 Example: Configuration file 'basesys.ini'

### Configuration file basesys.ini

The following defaults are applied to the template_basesys.ini file which is supplied in the / siemens/system/etc directory:

```
; ------------------------------------------------------------------
; Default Linux basesystem configuration
; ------------------------------------------------------------------
;
; section ID is for Windows compatibility and is ignored ;)


[ExternalInterface]
; If DisableCompanyNet is set to 1, the whole interface will be
disabled.
;DisableCompanyNet=1


; If ExternalIP is set, you can force the external Ethernet
; interface to use a fixed IP address etc. instead of using DHCP
; If a Hostname is set, it even overrides one received by DHCP
; Please note that only letters, digits and '-' are allowed in
Hostname;
```

```
; specifically '_' or '.' are forbidden!
;ExternalIP=210.210.210.210
;ExternalNetMask=255.255.255.0
;Gateway=210.210.210.1
;Nameservers=210.210.210.1 210.210.210.2
;Timeservers=210.210.210.3
;Hostname=somename
;Domain=example.com

; If ExternalIP is empty (default), DHCP is used with the following
; ClientID the default is "@MAC" to use the MAC address,
; alternatives are "@NAME" to use the hostname (Hostname above),
; "@CFID" to use the ID of the CF card if one is available
; (like on X120), or any other arbitrary string
;DHCPClientID=@MAC

[InternalInterface]
; If DisableSysNet is set to 1, the whole interface will
; be disabled
;DisableSysNet=1

; With InternalIP and InternalNetMask (both must be set together),
; you can change the address on the internal/TCU/automation/system
net.
;InternalIP=192.168.214.1
;InternalNetMask=255.255.255.0

; These two can define an alias IP for X120.
;InternalIP_Alias=192.168.216.1
;InternalNetMask_Alias=255.255.255.0

; This defines a "name of station" for ProfiNet (ERTEC interface)
; for use with DCP, default is the normal hostname.
; Please be careful if a PN-Name can also be set by other means,
; for example by NCU-Link. Usually it is not needed to define
```

```
; a name here.
;PN_StationName="PN_IO"

; Setting EnableDHCPD_SysNet to 0 suppresses that a DHCP server is
; started on that interface
;EnableDHCPD_SysNet=0

; Set synchronization of all DHCP servers in the
; system/TCU network (X120). Possible values are:
; OFF,
; or ON_LOW (low priority),
; or ON or ON_HIGH (normal priority),
; or ON_MASTER (highest priority),
; or ON_CLIENT_SYNC (starts DHCP client only, synchronization
active),
; or ON_CLIENT_NO_SYNC (starts DHCP client only, synchronization
inactive).
; ON_MASTER is used to make this machine deterministically the
; DHCP master server, but should be used only for one server
; in the network.
;SyncModeDHCPD_SysNet=ON

; This is the range of dynamic IPs given out by the DHCP server
; Defaults should be sensible
InternalDynRangeStart=192.168.214.10
InternalDynRangeEnd=192.168.214.239

; If once a DHCP master was seen, the DHCP server waits this many
; seconds for the master to show up before trying to become the
; active server
;DHCPDNoMasterWait=120

; DNS domain name used for names in system/TCU network
; (default is "local" to conform with RFC)
InternalDNSDomain=local
```

```
; FixedDomain and FixedNameservers can be used to define a DNS domain
; and nameservers, if the DHCP/DNS server on system network
; is disabled.
;FixedDomain=local
;FixedNameservers=192.168.214.1

; If DisableNATRouting is set to 1, the NCU won't forward
; from TCU or IBN net
; (X120 anad X127, resp.) to external net (X130).
; If set to "X120" or "X127", forwarding will only be
; disabled from that interface.
;DisableNATRouting=1

[IBNInterface]
; If DisableIBNNet is set to 1, the whole interface will be disabled
;DisableIBNNet=1

; Setting EnableDHCPD_IBNNet to 0 suppresses that a DHCP server
; is started on that interface
;EnableDHCPD_IBNNet=0

; If DisableIBNForwarding is set to 1, the NCU won't do NAT routing
; from IBN network (X127) to system network (X120).
;DisableIBNForwarding=1

; If EnableSysNetToIBNForwarding is set to 1, the NCU forwards
; packets from X120 to X127. (Please note that the sending host must
be
; able to route the packets to the NCU by its own means, the NCU just
; accepts and forwards packets to 192.168.215.x with this setting.)
;EnableSysNetToIBNForwarding=1

[SNMP]
; The following variables define strings which are delivered on
; certain SNMP requests.
```

```
; SNMPLocation and SNMPContact are reported for the standard OIDs
; SNMPv2-MIB::sysLocation and SNMPv2-MIB::sysContact, resp.
; SNMPAutLocation and SNMPFunction are used in the Siemens A&D
specific
; AUTOMATION-SYSTEM-MIB as
automationSystemIdent.automationLocationTag
; and automationSystemIdent.automationFunctionTag, resp.
SNMPLocation="not specified"
SNMPContact="not specified"
SNMPAutLocation="not specified"
SNMPFunction="not specified"
SNMPStation=""

[DCP]
; These variables en/disable the DCP protocol on X120/X127
; and X130, resp.
;InternalDcpEnabled=0
;ExternalDcpEnabled=1

[LLDP]
; These variables en/disable the LLDP protocol on X120/X127
; and X130, resp.
;InternalLldpEnabled=0
;ExternalLldpEnabled=1

[LinuxBase]
; Keep size of /var/log/messages around this value
; (not followed exactly for performance reasons)
LogfileLimit=102400

; Protocol/Port pairs to open in the firewall
; (e.g. TCP/5900, UDP/514, ...)
;FirewallOpenPorts=TCP/5900 TCP/102 TCP/22

; If there is only one Ethernet interface, it's used by
; default as an external (company) network. Alternatively,
```

```
; usage as automation net (TCU boot support etc.) is possible
; by setting NetworkModel to "automation". (NCU only, Service
; and PCU20A variants!)
;NetworkModel=automation

; DisableSubsystems can be used to skip certain CFSes (= subsystems)
; a list of multiple names (separated by spaces) is possible
; a simple name means all CFSes with this name, an absolute path
; (e.g.,/siemens/sinumerik/nck) exactly this one CFS with
; the special value "ALL", all subsystems can be disabled
;DisableSubsystems=nck

; Directory for OEM netboot files (served by tftp to diskless clients)
NetbootDirectory=/oem/common/netboot

; Default security type for all CIFS mounts, if not given explicitly
; by sc connect -sec=TYPE. If not defined, NTLMv1 is used or NTLMv2
; if the server allows.
;DefaultCIFSSecurity=ntlmssp

; Properties of local time zone: names, offset, start and end day
; Some examples:
;  Europe: WET0WEST,M3.5.0,M10.5.0
;          CET-1CEST,M3.5.0,M10.5.0
;          EET-2EEST,M3.5.0,M10.5.0
;   USA:   EST5EDT,M4.1.0,M10.5.0
;          CST6CDT,M4.1.0,M10.5.0
;          MST7MDT,M4.1.0,M10.5.0
;          PST8PDT,M4.1.0,M10.5.0
;   China: CST-8
;   Japan: JST-9
Timezone=UTC
; TimezoneName(for-into-only-do-not-change)=Etc/Universal

; If DisablePLCTimeSync is set, no time synchronization with
```

```
; PLC will happen
;DisablePLCTimeSync=1

; PLCTimeType informs time synchronization what kind of time
; is kept in PLC:
; local: PLC uses "wall clock time", with switches between DST and
;   standard time, if DST exists in the selected Timezone
;   If Linux synchronizes to an external server and passes time on
;   to PLC, this means the PLC clock is changed twice a year.
;   For PLC time-of-day interrupts this means they can be skipped
;   (begin of DST) or executed twice (end of DST). If this might be
;   an issue,try to avoid such interrupts between 2am and 3am.
;   If Linux receives time from PLC, it assumes that this clock will
;   switch between DST and standard time somehow (manually or
;   automatically). If this doesn't happen (or is delayed) Linux time
;   will be 1h behind the correct value. NCK and drives will still
;   follow PLC clock.
;
; local-nodst: PLC uses local time as above, but without DST
;   (summertime). All components (PLC, NCK, HMI, drives) always
;   use standard time, even if the selected timezone has a DST.
;   If Linux receives time from the PLC, it assumes standard
;   (non-DST) time, too. If this is not the case, the Linux clock
;   will be 1h ahead of the correct value.
;   This time type has been used implicitly up to SW 4.7.3 and is
;   still the default to maintain compatibility.
;
; utc: PLC clock contains universal time (= GMT+0)
;   All components (PLC, NCK, HMI, drives) will use UTC.
;   However, S7-300 engineering and HMI Operate 4.7 are not prepared
;   for displaying UTC times as local times yet.
;   This type makes synchronization simple, as no changes for
;   timezone or DST are needed. It also avoids changes to PLC clock
;   and supports applications across different timezones.
;   If Linux receives time from the PLC, it assumes UTC there.
```

```
;   If this is not the case, the Linux clock will have a permanent
;   offset according to local timezone.
;PLCTimeType=local

; Set to 1 to enable coredumps in /var/tmp
EnableCoreDumps=0
```

# 2.4 Service Commands

## 2.4.1 Using service commands

### Overview

The service command 'sc' is a tool used for performing a range of service tasks on a SINUMERIK NCU. The required action is written in the command line after the 'sc', e.g.: sc help

This action produces a list of all actions with a short description. After the action, further parameters or options can follow.

'sc' is available in both the NCU basic system and the service system (mostly in the form of a USB memory.) However, some actions are only useful in one of the two systems (NCU/ Service). This is noted for each action.

### See also

The most important terms and abbreviations are explained in the glossary.

### Privilege levels

The 'sc' command can execute its actions using more privileges than are normally granted to the calling user. For example, starting or stopping subsystems requires root privileges, but 'sc' allows every user who belongs to the ‘service’ group to do this.

Each action of 'sc' is assigned a “privilege level”. This is a user group to which the user must belong in order to execute the action. As the groups are hierarchically nested, members of “higher” groups can also use the action in question. For example, the group 'manufact' is above 'service', which means that members of the group 'manufact' can call all actions that require the privilege level 'service'.

The privilege level that the caller requires is noted for each individual action. The possible levels are (in ascending order):

- none
- operator
- user
- service
- manufact

**Example:**

An action with the privilege level 'user' can also always be executed by members of the groups 'service' and 'manufact'. Actions with privilege level ‘none’ can be called by all users.

If a user does not have the required privileges, the following error message is output:

```
Action 'ACTION' needs at least GROUP privilege level.
```

## 2.4.2 Syntax for the Actions

### Description

'sc' in the command line is essentially not case-sensitive.

The following entries are therefore interpreted as identical:

**sc help show**

**SC help SHOW**

**sc HeLp sHoW**

In some cases, however, upper/lower case can make a difference, for example in file or user names. This is avoided wherever possible.

The conventions used are as follows:

- Names completely in upper case represent objects to be used depending on the situation.
  Example: `sc help ACTION`
  In this case, ACTION is to be replaced by the action for which you want a description. If written in lower case, however, the entry should be entered as specified.
- Square brackets indicate optional entries.
  Example: `sc help [ACTION]`
  In this case the specification of an action is optional, which means that you can enter an action, but do not have to. Square brackets may also be nested:
  ... [USERNAME[/DOMAIN]] ...
  In this case, USERNAME and DOMAIN are both optional, but you can only enter a DOMAIN if you have also entered the USERNAME.
- Alternatives are separated with '|'.
  Example: `sc start all|system|SUBSYSTEM`
  This means that any of the following commands applies:
  ```
  sc start all
  sc start system
  sc start SUBSYSTEM
  ```
  In the latter case, SUBSYSTEM in upper case can be replaced by a concrete subsystem name.
- As a shortened form, alternatives can also be written in square brackets:
  `sc save [-full|-user] ...`
  Here you can use the option "-full" or "-user", or none at all.
- Options that begin with '-' can always be entered in any order.
  For example, the notation could be interpreted to mean that "-force" must come after "-full" or "-user", but this is not necessarily the case:
  `sc save [-full|-user] [-force] FILENAME`

## 2.4.3 Generic elements

### Overview

This sections describes the syntax elements that are used by several actions.

### Permitted interface designations

The names of network interfaces are used by "sc show ip" and "sc enable DHCPSvr", for example.

As the input of an interface is optional, it is always introduced by a '−' character. The '−' is followed by the actual name. In most cases, multiple names are possible for the same interface.

Accepted names are:

System network port: "X120", "eth2", "tcu", "internal"
Company network port: "X130", "eth1", "factory", "extern
IBN port: "X127", "ibn0", "pg"

### Permitted subsystem designations

Subsystem names are entered when using "sc enable" and "sc start", for example. In most cases, the name of the subsystem is simply the name of the corresponding CFS, without the path and without the extension ".cfs". For example, for the CFS /siemens/sinumerik/nck.cfs, the relevant subsystem name is simply "nck".

It is also possible to use absolute paths (beginning with /) in a subsystem name. In the example above, you could also use "/siemens/sinumerik/nck" as a subsystem name. The difference between names with and without a specified path is that without the path, all CFS with this name are included, but if the path is specified, only this exact CFS is meant.

### Subsystem

A subsystem is a CFS that not only contains a collection of files, but also executes a program, for example, at runtime. To do this, the CFS contains a script that is used to control the starting and stopping of this program.

For this reason, only administrators are permitted to set up NFS file systems, and NFS is usually only implemented in uniformly administrated environments. Exported file systems on the server are addressed directly on the server via their path.

### CFS (Compressed File System)

A CFS (file extension ".cfs") is a compressed file system, similar to a zip file. It contains files and subdirectories that look like normal files on the controller at runtime. Files and directories contained in a CFS cannot be changed. They are decompressed at runtime as required.

### NFS (Network File System)

NFS is the most common protocol for remote file systems in the world of Unix, and is also available for Windows. NFS is closely based on the Unix privilege model – each time a file is accessed, a UID and GID are supplied which the server then uses to decide whether the operation is permitted. The server relies on the client to provide the correct IDs.

### Remote File System

A file system that is contacted over the network. The files are physically located on another computer in the network (the "server"), but appear locally the same as all other files. Operations performed on these files are sent via the network to the server, instead of being executed directly on a local storage medium (such as a hard drive or CompactFlash Card).

As a server usually exports more than one file system, a name for the required file system must also be entered in addition to the name of the server.

### SMB (Server Message Block)

SMB is the underlying protocol of MS Windows file systems (also known as drives, releases, shares, etc.). SMB connections are always active in the context of a specific user, who must be known to the server. Exported file systems have a name (release name), by which they can be addressed. The client does not need to know the concrete path on the server.

# 2.5 Description of the actions

## 2.5.1 Help

### Description

| | |
|---|---|
| Syntax: | sc help [Action] |
| Alternative names: | -h, - -help |
| Authorization level: | None |

Calling "sc help" without any additional action outputs a list of all possible actions with a brief description. If you enter an additional action, you receive a more detailed description for this action.

**Examples**:

```
sc help
All actions:
help [ACTION]
Print help about a specific or list all actions
restart
Reboot the machine
enable hmi|nck|SUBSYSTEM...
enable DHCPSvr -INTERFACE
Enable HMI, NCK, or any other subsystem
[...]
sc help enable
enable hmi|nck|SUBSYSTEM...
DHCPSvr -INTERFACE

Enable subsystem(s), like 'hmi', 'nck', and so on. A subsystem name
is the name of the CFS containing it, without the '.cfs' extension.
This enables all CFSes with that name, but you can also use a full
path (e.g., /siemens/sinumerik/nck) to enable just a specific CFS.
Another form is to enable the DHCP server on a network interface,
for example 'enable DHCPSvr -X120'.
```

## 2.5.2 Check-cf

### Description

| | |
|---|---|
| Syntax: | sc check-cf |
| Alternative names: | checkcf |
| Authorization level: | user |

With this action, the CompactFlash Card is read to check whether it has defective sectors. If errors occur, this is noted in the /var/log/messages file.

## 2.5.3 Clear

### Description

Various actions can be executed using command "sc clear".

### sc clear dhcp

| | |
|---|---|
| Syntax: | sc clear dhcp [-INTERFACE] |
| Alternative names: | --- |
| Authorization level: | service |

This command clears any state of the DHCP server at the specified interface (default setting is the system network) and resets it to its initial state. This means that the Lease data is deleted: All IP addresses are re-assigned in the network and the server "forgets" having ever detected a master server.

This action is only appropriate on a machine on which the active DHCP server runs.

**Note**

The clearing of the Lease data does not delete the entire file, but only the data contained therein. This action also increments the version number, so that available standby DHCP servers can also perform the deletion.

### sc clear preinstalled-keys

| | |
|---|---|
| Syntax: | sc clear preinstalled-keys |
| Alternative names: | --- |
| Authorization level: | service |

This command deletes all of the SSH keys preinstalled by Siemens on the control system. When called from the service system, the key on the CompactFlash Card is involved, and not the SSH keys on the service system itself.

| NOTICE |
| --- |
| **Accessing the control** |
| When removing the SSH key preinstalled by Siemens, then the risk of unauthorized access using it is eliminated, assuming this is known. |
| However, in order to ensure sufficient access to the system, before deleting we recommend that a new SSH key is defined and installed. |

## 2.5.4 Closeport

### Description

| | |
| --- | --- |
| Syntax: | sc closeport ID |
| Alternative names: | --- |
| Authorization level: | service |

### Principle of operation

This command closes a port in the firewall opened previously with the "sc openport" command. This is only necessary when the port is to be closed manually before the time set with "sc openport" expires. Otherwise the port is closed automatically when the port service life expires.

One parameter is the ID number of the firewall rule output by "sc openport".

## 2.5.5 Connect

### Description

| | |
| --- | --- |
| Syntax: | sc connect [-ro] [-sec] SERVER:/PATH [MOUNTPOINT]<br>sc connect [-ro] [-public] [-sec] //[USERNAME[/DOMAIN] [%PASSWORD]@]SERVER/SHARE [MOUNTPOINT] |
| Alternative names: | mount |
| Authorization level: | none |

This action makes a remote file system on a server available on the control. This is enabled by linking the remote file system to a local directory, known as the "MOUNTPOINT". The files offered by the server are then visible under this directory.

| NOTICE |
| --- |
| **Password protection**<br>Note that when entering this command, the password appears on the screen in plain text according to the specified syntax.<br>For path names, use the slash “/” and not the backslash “\”. |

## Supported file systems

Two types of remote file system are supported: Windows SMB and Unix NFS. These two systems have completely different characteristics, particularly in terms of user administration:

- In Windows SMB, you connect to the server as a particular user that the server must recognize. Via this connection, you then access the files as this user, independently of which local user triggers the action.
  This feature means that in SMB systems, you already have to enter a user name, if necessary its domain, and a password when the connection is established.
- For NFS, the connection itself does not require a particular user to be entered. Instead, for each file operation, the user who wants to carry out the operation must log on to the server. The server then decides whether or not to permit this. Users are entered using a user ID and group ID, not using names. The server must therefore recognize the corresponding IDs (or permit access for all users).
  Another type of remote file system that is supported are USB storage media exported from the TCU (USB Flash Drive). Since these are integrated using NFS, the entries for server and path are similar to those for NFS. However, the TCU names are administrated differently, and the USB storage media have specific paths that do not physically exist.

## Authentication using "security type"

Various Windows servers have requirements regarding the authentication technique used. If the server only accepts certain types, and the default setting is not appropriate, then for CIFS and SMB, a security type can be specified using option -sec.

Permissible values include:

- ntlm (= NTLMv1; still available only for compatibility reasons)
- ntlmv2
- ntlmv2i
- ntlmssp
- ntlmsspi

The "i" attached to the end, forces that the communication is signed.

Without the -sec option, the value of variable "DefaultCIFSSecurity" is used in basesys.ini (Page 25). If this is also not defined, then the default setting is NTLMv1 or NTLMv2.

| NOTICE |
| --- |
| **Authentication**<br>It is recommended that NTLMv1 is no longer used, as the security associated with this technique is no longer guaranteed. |

## Notation of the remote file system

For SMB and NFS/TCU, the file system is entered using different notations:

### SMB: //[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE

The fixed share is: //SERVER/SHARE

The server name can of course also be a numeric IP address. SHARE is the name of the release on the server. Note that the character '$', which often occurs in this type of name, must be preceded by a backslash ('\') in the command line. Otherwise, the system tries to expand a variable.

A user name can also be entered in front of the server name, separated with a '@' character. If necessary, the user name can also be extended by adding '/' followed by the Windows domain to which it belongs. The password belonging to this user is normally queried interactively, so that it is not visible on the screen.

For some applications, however, it may be necessary to write the password in the command line. (For example, programs started from WinSCP cannot read from the keyboard.) In this case, you can append an additional '%' character, followed by the password. If this contains any special characters that are interpreted by the shell (<, >, &, ;, ", ', $, (, ), |), you should shield these by preceding them with a backslash. Commas in SMB passwords cannot be interpreted.

### NFS: SERVER:/PATH

In NFS, entering the user is not required, which just leaves the two components SERVER and PATH. As before, SERVER is the name of the file server. In contrast to SMB, PATH is a path name that exists on the server, and not an arbitrarily assigned name.

The command "sc show drives SERVER" shows the file systems offered by a particular server in the correct notation. For SMB, if required, the user name, etc. still need to be added.

After the entry of the remote file system, you can also specify the required directory ('MOUNTPOINT') in which the remote files will become visible locally. This should be an empty directory to which you have write access. If MOUNTPOINT is omitted, 'sc' itself generates a suitable directory. The name is /tmp/drvNN (numbered consecutively), and is displayed when the connection is successful.

The "-ro" option connects the remote file system in read-only mode, this means that you can only read and do not have write access. This can be useful if you want to exclude changes that originate from the control, or if the server only allows read-only connections.

As explained above, an SMB file system is connected as the specified user. To prevent a third user from executing file operations on the control in the name of the connected user, the local directory can normally only be accessed by the latter (and all members of the same group).

However, if you do want to make the remote directory available to further local users, you can use the "-public" option to make the MOUNTPOINT belong to the "operator" group, and this access is therefore guaranteed.

## 2.5.6 Disable

### Description

| | |
|---|---|
| Syntax: | sc disable hmi\|nck\|SUBSYSTEM ... |
| | sc disable DHCPSvr -INTERFACE |
| | sc disable DHCPSync [-X120] |
| | sc disable IPAlias [-X120] |
| | sc disable usb [-MINUTES] all \| HOSTS |
| Alternative names: | --- |
| Authorization level: | service |

The "sc disable" command switches the specified subsystems off. These subsystems are then not loaded again at the next cold restart. You can also switch several subsystems off with a single "disable" command. "hmi", "nck" or any other subsystem names can be used for the subsystems.

**Note**

The "sc disable" command acts on the parameters in the basesys.ini file located in the card/user/system/etc/basesys.ini directory.

"sc disable IPAlias" deactivates the second alias IP address for the X120 network interface.

### See also

"InternalIP_Alias" and "InternalNetMask_Alias" parameters in the "basesys.ini" file.

### Special subsystems

The "DHCPSvr" name is a special case. This does not switch a normal subsystem on or off, but specifies whether or not a DHCP server should be started on a network interface. The interface is entered in the standard form.

The synchronization of the DHCP server is switched on and off with "DHCPSync". The synchronization is only possible in the system network.

The "sc enable" and "sc disable" commands function by changing certain variables in the /user/system/etc/basesys.ini file. The associated variable is "DisableSubsystems" or "DisableDHCPD<INTERFACE>" for subsystems and DHCPSvr, respectively.

"sc disable usb" locks USB storage devices: Specifically, the devices of USB class "storage" are involved, i.e. input devices can still be operated at the USB interface.

- Without a time specification [-MINUTES] , a permanent disable is set. When a time is specified, the inhibit is specified in minutes.
- The inhibit can refer to all network participants of the system network or a list with host names or IP addresses can be specified, for which the inhibit is valid.

## 2.5.7 Disconnect

### Description

| | |
|---|---|
| Syntax: | sc disconnect MOUNTPOINT |
| | sc disconnect all |
| Alternative names: | umount, unmount |
| Authorization level: | user |

The "sc disconnect" command terminates the connection to a remote file system previously connected using "sc connect". If the MOUNTPOINT was created automatically by 'sc', the associated directory is also deleted again.

The "sc disconnect -all" variant can be used to remove all existing SMB and NFS file systems.

## 2.5.8 Distribute

### Description

| | |
|---|---|
| Syntax: | sc distribute [parameter] |
| Alternative names: | dist |
| Authorization level: | service |

This command distributes TCU data to other machines in the system network. Only the "tcudata" parameter is relevant here.

#### distribute tcudata

| | |
|---|---|
| Syntax: | sc distribute tcudata |
| Alternative names: | --- |
| Authorization level: | service |

This command informs the active DHCP server that TCU data in /user/common/tcu has been manually changed. The server can then distribute this data to the standby servers.

It is recommended that the version file in /user/common/tcu be changed manually as this is only read-in once by the DHCP server at the start. An incrementing of the version would then only take effect at a cold restart.

**Note**

This command can be performed on any machine in the system network, not only on the active server. A message is always sent to the active server.

## 2.5.9 Enable

### Description

| | |
|---|---|
| Syntax: | sc enable hmi\|nck\|SUBSYSTEM ... |
| | sc enable DHCPSvr -INTERFACE |
| | sc enable DHCPSync [-X120] [-PRIORITY] |
| | sc enable IPAlias [-X120] IPADDR[/NETMASK] |
| | sc enable usb [-MINUTES] all \| HOSTS |
| Alternative names: | --- |
| Authorization level: | service |

The "sc enable" command switches on the specified subsystems such that these are also activated when a subsystem is started. You can also switch several subsystems on with a single "enable" command. "hmi", "nck" or any other subsystem names can be used for the subsystems.

**Note**

The "sc enable" command acts on the parameters in the basesys.ini file located in the card/user/system/etc/basesys.ini directory.

"sc enable IPAlias" defines a second alias IP address for a network interface. This is only supported for X120. The IP address itself has the usual notation with four numbers separated by decimal points. If not specified, the subnet mask can be determined from the class (A/B/C) of the IP or determined explicitly. The CIDR notation (number of bits of the network section) is also possible.

### See also

"InternalIP_Alias" and "InternalNetMask_Alias" parameters in the "basesys.ini" file.

### Special subsystems

The "DHCPSvr" name is a special case. This does not switch a normal subsystem on or off, but specifies whether or not a DHCP server should be started on a network interface. The interface is entered in the standard form.

The "sc enable" and "sc disable" commands function by changing certain variables in the /user/system/etc/basesys.ini file. The associated variable is "DisableSubsystems" or "DisableDHCPD<INTERFACE>" for subsystems and DHCPSvr, respectively.

With "sc enable usb" USB storage devices on permanently disabled USB interfaces or on certain host names or IP addresses are enabled again.

### DHCP synchronization

The synchronization of the DHCP server is switched on and off with "DHCPSync". The synchronization is only possible in the system network (X120). A priority can also be specified: -LOW, -HIGH or -MASTER.

The priorities have the following effect for the synchronization of the DHCP server:

- **MASTER:** The computer node becomes an active DHCP server. If several servers have been configured in the system network, this computer has the highest priority.
- **HIGH:** The computer node belongs to the server candidates with high priority, i.e. if no server with "MASTER" priority becomes active, then a computer with "HIGH" priority can be the active server.
- **LOW:** The computer node belongs to the server candidates with low priority, i.e. if no server with "MASTER" priority or "HIGH" priority becomes active, then a computer with "LOW" priority can be the active server.

**Note**

Recommended settings are:

- DHCP operation and DHCP synchronization are switched on in the system network.
- **Exactly one** NCU is set as DHCP server.
- **A maximum of two** computers are candidates with "HIGH" priority.
- All other components are set as DHCP clients or candidates with "LOW" priority.

### See also

"SyncModeDHCPD_SysNet" parameter in the "basesys.ini" file.

## 2.5.10 ntpdiag (diagnostics)

### Description

| | |
|---|---|
| Syntax: | sc ntpdiag |
| Alternative names: | --- |
| Authorization level: | None |

Using command "sc ntpdiag" , a list of the configured time servers and their status is output.

In addition, for the time server, which the NCU uses, information such as IP address, stratum (distance to a time character receiver, e.g. an atomic clock) and the offset between the NTP

server and its client are output. If the connection to the server is poor, a ping command is issued in order to check the general accessibility situation.

## 2.5.11 Openport

### Description

| | |
|---|---|
| Syntax: | sc openport [-MINUTES] PROTO/PORT SOURCE ... |
| Alternative names: | --- |
| Authorization level: | service |

### Principle of operation

This command opens a port in the firewall to the company network (X130) for a certain time. The default time is 15 minutes, but this can be changed with the -MINUTES option. The maximum possible time is 60 minutes.

The port to be opened is specified in the form "PROTO/PORTNR". The protocol can be either "tcp" or "udp". This is followed by the specification of the hosts from which the port should be accessible. Several forms are possible here:

- A single IP address: "128.128.12.12"
- A host name (if this can be resolved via DNS), e.g. "server"
- An IP area with net mask specification, e.g. "128.128.12.0/255.255.255.0"
- An IP area with specification of the valid bits, e.g. "128.128.12.0/24"

If the specification of the source host is completely missing, this is normally acknowledged with an error message. However, an exception is when the "sc openport" command is entered in a shell opened via SSH. The permitted host is taken from the $SSH_CLIENT environment variable that contains the IP of the SSH client.

If the opening was successful, an ID number is output in a status message:

```
sc: Port tcp/25 is open, rule ID is 6620
```

This ID can be used with "sc closeport" to manually close the port earlier.

| NOTICE |
|---|
| **Security risk** |
| Note that opening the ports in the firewall can represent a security risk. Only enable the ports that you actually require. |
| Note that every accessible service can have security gaps. |

### See also

"FirewallOpenPorts" parameter in the "basesys.ini" file.

## 2.5.12 Portstatus

### Description

| | |
|---|---|
| Syntax: | sc portstatus [-x127] PORTSPEC SOURCE |
| Alternative names: | --- |
| Authorization level: | service |

### Principle of operation

The "sc portstatus" command queries whether a specific port in the firewall is open. The "-x127" option must be specified if X127 rather than the X130 firewall is concerned. The port specification is identical to "sc openport", e.g. "tcp/102". However, no names are permitted here and the specified source IP must be unique.

The following status is output:

- "Port not open" with exit status 1.
- "Port is open (rule ID is <N>)." and exit status 0.
  Whereby, the ID required for "sc closeport" must also be specified.

## 2.5.13 Restart

### Description

| | |
|---|---|
| Syntax: | sc restart all\|system\|network\| [!]SUBSYSTEM |
| Alternative names: | reboot |
| Authorization level: | service |

The "sc restart" action initially stops the specified subsystems and then restarts them with the following restriction: Dependent subsystems are also started for a restart.

### Syntax

'hmi', 'nck' or any other subsystem names can be used as subsystems. Multiple subsystems can also be specified successively; they (including all required dependencies) are then started and stopped in this order. It is also possible to start and stop all subsystems, other than the named ones. To do this, prefix an exclamation mark "!" to the list.

"all", "system" and "network" are special values for the subsystem:

- "all" stops and starts all available subsystems.
- In contrast, "sc restart system" initiates a cold restart of the system (reboot).
- "network" may be specified only together with "restart". "sc restart network" reinitializes the network, e.g. after settings have been changed in basesys.ini.

### Examples

# Stop all subsystems:

$ sc stop all

# Stop only HMI:

$ sc stop hmi

# Start HMI and NCK:

$ sc start hmi nck

# Start all subsystems other than NCK:

$ sc start ! nck

## 2.5.14 Restore

### Description

| | |
|---|---|
| Syntax: | sc restore [-full\|-addon\|-addon+\|-oem\|-oem+\|-user] [-force] [-nodelete] [-update] [-restart] FILENAME |
| Alternative names: | --- |
| Authorization level: | user |

A backup created with "sc save" can be restored to the control with "sc restore".

### Principle of operation

As for "sc save", a restore is possible only from the service system or when the subsystems are stopped. If "sc restore" reports an error, for example that the restore will cause running subsystems to crash, you can use the "sc save" option to force the operation, as for "-force".

Normally, "sc restore" deletes the complete target area before the backup is restored (complete CompactFlash Card for full backup, /user in user data backups). This means that no files are subsequently available that were not included in the backup.

### Archive and backup files larger than 4 GB

If a backup consisting of several archive files is to be restored with "sc restore", only the first file of the backup must be specified in the FILENAME parameter. Once the end of the file has been reached, other parts will be searched automatically.

Decisive when determining the continuation files is the format of the file name that was transferred during calling, i.e. when calling "sc restore /tmp/backup.01of02.tgz", and after reaching the end of "/tmp/backup.01of02.tgz", a search is next made for the "/tmp/backup.02of02.tgz" file. When calling "sc restore /tmp/backup.1of2.tgz", the "/tmp/backup.2of2.tgz file is expected correspondingly as a second part of the backup.

### Options

If no further options are specified, the archive should represent a full backup and this should be restored in full. The status of all files is therefore the same after the "restore" as it was at the time of the backup.

- The -full option additionally forces the partition and the file system to be recreated on the CompactFlash Card. This is, however, possible only from a service system. -full is necessary if the partition table and/or file system is missing or damaged.
- However, if you want to restore the backup files without losing any files that have been created in the meantime, you can use the "-nodelete" option to prevent them from being deleted. "-nodelete" is not executed together with "-full", because all data is always deleted when the file system is recreated.
- The -addon, -oem, -user, -addon+ and -oem+ options also allow only parts of an archive to be unpacked.
- The -update option is used to load software updates supplied by Siemens. The -update option can also be useful with -restart. If user-defined system files are changed when loading the archive to the control, a cold restart or reinitialization is necessary. In such cases, "sc restore" outputs an appropriate message at the end of the operation. The -restart option causes the required action to be performed automatically.

## 2.5.15 Save

### Description

| | |
|---|---|
| Syntax: | sc save [-full\|-addon\|-addon+\|-oem\|-oem+\|-user] [-force] [-update] FILENAME |
| Alternative names: | backup |
| Authorization level: | user |

The "save" or "backup" action creates a backup of the CompactFlash Card in FILENAME, which specifies the archive path and also the file name. If the action is used from a service system, the backup receives the files of the associated control – and not the service system.

**Example:**

```
sc save -user /tmp/drv01/backup.tgz
```

### Options

The following options are used for selecting which directories of the file system are to be written to the archive:

| | |
|---|---|
| -full: | complete backup (default): all files of the control, incl. Boot Loader |
| -addon: | only /addon directory |
| -oem: | only /oem directory |
| -user: | only /user directory (user data) |

-addon+: /addon, /oem and /user

-oem+: /oem and /user

If "sc save" is used directly on the control, and the subsystems are running, this can lead to inconsistencies between saved files because they could still change during the backup process. Therefore, in this case "sc save" exits with the message that subsystems are still running. If you want to create a backup anyway, you can use the -force option. The message to the subsystems is still output; however, the backup operation still continues.

If "sc save" is started from the service system, assuredly no subsystems of the control are running and "-force" is not necessary.

### Archiving and backup files larger than 4 GB

If the backup is written to an FAT32 target file system, which does not permit any file size greater than 4 GB, the backup is divided into several files, each of which is smaller than this threshold value.

In the target file name that is specified when "sc save" is called, in this case, details of which part of a backup is involved and the total number of associated files are provided: Calling "sc save /tmp/backup.tgz" thus generates, for example, the "/tmp/backup.01of02.tgz" and "/tmp/backup.02of02.tgz" files.

## 2.5.16 save-nvram (data backup)

### Description

| | |
|---|---|
| Syntax: | sc save-nvram [-drive\|-plc\|-tco] [DIRECTORY] |
| Alternative names: | --- |
| Authorization level: | None |

Using command "sc set-nvram", the content is saved to a non-volatile memory (non-volatile memory). Depending on the machine type and equipping, such an NVRAM is provided in the integrated drive, in the PLC and as TCO-EEPROM.

Using options -drive, -plc and -tco, individual NVRAM types can also be explicitly selected for backup. Backing up all existing types is the default setting.

The content is written to files called "nvram-TYP.bin"; whereby, TYP is replaced by "drive", "plc", .... The files are created in the current directory if no directory is specified.

## 2.5.17 set-svc-route (gateway)

### Description

| | |
|---|---|
| Syntax: | sc set-svc-route GATEWAY DESTNET |
| Alternative names: | --- |
| Authorization level: | None |

A special network route can be entered using command "sc set-svc-route". A typical application is to make a service network, connected via modem, accessible, e.g. for remote diagnostics.

The parameters are the GATEWAY, via which the packets for DESTNET should be routed. The gateway must be specified with its IP address, and this must be located in a network that is **directly** connected. The gateway must be able to be accessed without additional intermediate routers.

| NOTICE |
|---|
| **Routing in the network**<br>The target network, which should be applicable for the detour via GATEWAY, can be specified in the usual syntax with network mask ("128.128.12.0/255.255.255.0") or CIDR ("128.128.12.0/24"). It is not permissible that it overlaps with one of the directly accessible networks of the NCU - neither completely nor partially. |

## 2.5.18 Show

### Description

Various displays regarding the system state are summarized in command "sc show".

### show ip

| | |
|---|---|
| Syntax: | sc show ip [-INTERFACE] |
| Alternative names: | ---- |
| Authorization level: | None |

This command displays the IP address data of network interfaces. You have the option to enter a specific interface. If no particular interface is specified, the data for all existing interfaces is displayed, as well as the Default Gateway.

**Example:**

```
sc show ip
X120 (system network, eth0):
   configured: (default)
   current : IP=192.168.214.1 Netmask=255.255.255.0
MAC=08:00:06:73:55:fd
   DNS Name  : ncu1.local
```

```
   Nameserver: 127.0.0.1
   DNS Suffix: local
   DHCP      : synced server, prio=high, active
   Statistics: RX=0.0MB (0.00% errors), TX=0.2MB (0.00% errors)
X130 (company network, eth1):
   configured: DHCP
   current   : IP=111.222.333.64 Netmask=255.255.248.0
MAC=08:00:06:73:55:fe
   DNS Name  : name.test.siemens.de
   Nameserver: 111.222.333.12 111.222.333.13 111.222.333.14
   DNS Suffix: test.siemens.de
   DHCP      : client (server: 111.222.333.221)
   Statistics: RX=1.2MB (0.00% errors), TX=0.0MB (0.00% errors)
X127 (engineering network, ibn0):
   current   : IP=192.168.215.1 Netmask=255.255.255.224
MAC=08:00:06:73:55:ff
   DNS Name : ncu-ibn
   DHCP      : server
   Statistics: RX=0.0MB (0.00% errors), TX=0.0MB (0.00% errors)
Default gateway: 111.222.333.1 (via eth1)
Used nameserver: 127.0.0.1
Used DNS suffix: test.siemens.de local
```

The following data is displayed for each interface:

- **Name:** The name of the connection socket (X1_ _) together with the name used by the operating system in brackets (ethN or ibnN).
- **"configured":** The IP address configured in the basesys.ini (variables ExternalIP/ ExternalNetMask for the X130, InternalIP/Internal-NetMask for the X120), or "(default)" if nothing is configured in the basesys.ini, or "DHCP" if the address was sourced via DHCP.
- **"current":** The currently set IP address together with the network screen form and MAC address of the interface.
- **"DNS name":** The result of a DNS reverse lookup for the current IP address.
- **"Name server":** The DNS servers associated with this interface (e.g. receive DHCP via this interface) is output in this list.
- **"DNS suffix":** The DNS search suffix associated with this interface.
- **"DHCP":** A flag that indicates whether a DHCP client or server is running for this interface. In the case of a client, the server where the IP address comes from is also displayed. A DHCP server on X120 can additionally be synchronized in the system network. Then the information is displayed whether the server is the active server or is in standby mode.
- **"Statistics":** The total data volume received or sent via this interface, and the percentage of faulty packets.

When all interfaces are displayed, the default gateway is also output, i.e. the address of a router to which all packets are sent that cannot reach their destination directly via a local interface.

The default gateway is therefore a piece of data that applies to all interfaces and of which there is only one instance. However, there is one interface via which it must be possible to address the default gateway. This is displayed in brackets after the address.

### show drives

| | |
|---|---|
| Syntax: | sc show drives SERVER |
| Authorization level: | None |

"sc show drives" shows the remote file systems available for a particular server. The server name SERVER can be an NFS server or a "TCU" that represents any USB media connected to a TCU. For details of possible server names and how to enter a user name (often required for SMB), see the description of the "sc connect" command.

### Examples

Notation:

```
sc show drives someuser/somedomain@somepc # Windows-Server
Password: *******
//somepc/C$
//somepc/D$
//somepc/images
sc show drives someserver # NFS-Server
someserver:/export/home1
someserver:/export/home2
sc show drives TCU # TCU USB-Medien
TCU1:/dev0-0
TCU2:/dev0-0
```

All the available remote file systems of the relevant server are listed in the notation that is also expected by "sc connect".

- In NFS file systems, the server name is always first, followed by a colon and then the export path.
- A USB storage medium on a TCU is a special form of NFS, and therefore also has the same notation as an NFS file system. The specified path exists - but not physically - on the TCU, but is converted there to USB by the NFS server.

### show net

| | |
|---|---|
| Syntax: | sc show net [-xml] [-hw\|-tco\|-sw\|-swfull\|-loc\|-panel\|<br>-dhcp\|-switch\|-all] [HOSTS...] |
| Authorization level: | None |

The "sc show net" command displays the devices in the system network and shows additional information about these devices.

Because both these functions rely on SNMP (Simple Network Management Protocol), only SNMP-capable devices will be found. SNMP as of version 2.6 is available for Linux-based devices, for all other devices (e.g. MCP) this depends on the associated software version. If a simple call is made without any additional options, a list of the devices found will appear showing the relevant IP address, DNS name (if known), and a short description (module name).

## Options

The various switches, which can also be combined, allow additional information to be output on the devices found:

- Unless additional options are involved, the output format for "sc show net" is designed to be displayed on the screen. The alternative XML format is preferable, however, if further machine-based processing is required. This can be selected via the -xml option (this option must be the first one!).
- If the command line contains nothing after the switches, "sc show net" will search the complete system network via broadcast for devices able to use SNMP. This search takes time, however, and in the case of larger systems, the volume of information output can be considerable. In view of this, a list of IP addresses or (DNS) names can be specified to ensure a better overview. This will limit the search process and only the named devices will be included.

| | |
|---|---|
| -hw | Information on hardware, such as the MLFB, serial number, hardware ID, hardware version, SRM version, manufacturer, and the serial number of the CompactFlash Card. Some devices have two of these hardware blocks (the second one being intended for an integrated device).<br>The PLC in the NCU and the TACO in the OP on an TCU are two such examples. |
| -tco | Operating state data: Switch-on counter, operating hours counter, quantity of data written to the CompactFlash Card, number of times maximum temperature exceeded. |
| -loc | Location data: Location, function, and contact details for the device (if configured there). |
| -sw | Displays brief details of the software installed, only the version of the overall status (or of the basic system if only this is installed). |
| -swfull | Displays all the installed software components. They are represented hierarchically in the form of a tree. The following is provided for each component: Information about the version, internal version (if different), target version (if different), version details, and the installation path. |
| -panel: | Displays panel-specific data: Size and color depth of the display, MCP/TCU/DCK/EKS index, and the VNC server currently represented |
| -dhcp | Data about a synchronizing DHCP server: Operating state (off/standby/active), priority, versions of the lease file and TCU data, and the range for dynamically assigned IP addresses |

| | |
|---|---|
| -switch | Displays the settings of rotary switches on the device, e.g. the NC and PLC switches of an NCU or the DIPFIX of an HT 8. |
| -all | Equivalent to all the above switches. |

**Note**

The data and information read out here are automatically written to the EUNA file (End User Notification Administration), and saved on the associated device.

## 2.5.19 Start, Stop

### Description

| | |
|---|---|
| Syntax: | sc start all\|system\|SUBSYSTEM... |
| | sc stop all\|system\|SUBSYSTEM... |
| Alternative names: | --- |
| Authorization level: | service |

These two actions start or stop individual or all subsystems. Since these actions have a direct influence on the runtime behavior, they are available only directly on the control, but not in the service system (exception: "sc stop system"). The subsystems of the associated control do not run here, and can therefore not be controlled.

**Note**

Only "all" and "system" function, but not arbitrary subsystems. Both of these actions have been implemented only to a limited extent.

### Subsystem names

"hmi", "nck" or any other names can be used for the subsystems. Multiple subsystems can also be specified successively; they are then started and stopped in this order.

"all" and "system" are special values for the subsystem:

- "all" starts or stops all available subsystems.
- "sc stop system" also stops all subsystems and the basic system with the control.
- "sc start system" is identical to "sc start all" and exists mainly for reasons of symmetry.

## 2.5.20 usb (connect device)

### Description

Preconditions:

- Hardware: TCU 30.3
- Software: If a USB device is to be used, which requires additional drivers or software, then these must be available on the NCU.

| | |
|---|---|
| Syntax: | sc usb attach\|export\|unexport SERVER BUSID |
| | sc usb detach PORT |
| | sc usb showdevs SERVER |
| | sc usb status |
| Alternative names: | --- |
| Authorization level: | None |

Using command "sc usb", any USB devices, with the exception of hubs and data storage media, which are connected to a TCU can be made available on an NCU.

For parameter SERVER (TCU, whose devices are to be used), symbolic host names (e.g. TCU1) - as well as also IP addresses (e.g. 192.168.214.10) – can be specified.

### Examples

Listing devices connected to TCU1:

```
# sc usb showdevs TCU1
                0bf8:100c
```

Exporting devices from TCU1:

```
# sc usb export TCU1 1-2.1
```

In the list, this device is marked as being exported:

```
# sc usb showdevs TCU1
                0bf8:100c
                [exported]
```

Attaching the device to the NCU:

```
# sc usb attach TCU1 1-2.1
```

The device is listed in the local status of the NCU:

```
# sc usb status
Port 00: Port in Use
TCU1:1-2.1
```

Detach the device from "sc usb status" using the port number:
```
# sc usb detach 00
```

Unexport device on the TCU:
```
# sc usb unexport TCU1 1-2.1
```

The device is no longer marked as exported:
```
# sc usb showdevs TCU1
                0bf8:100c
```

## 2.5.21 vncpwd (password)

### Description

The following actions are executed using command "sc vncpwd":

- sc vncpwd set to assign a password.
- sc vncpwd reset to delete a password.

| | |
|---|---|
| Syntax: | sc vncpwd set companynetwork\|systemnetwork |
| | sc vncpwd reset companynetwork\|systemnetwork |
| Alternative names: | --- |
| Authorization level: | None |

Using command "sc vncpwd set", on the server (NCU) a password is set up for the VNC connection via the company network or the factory network. As soon as a password has been assigned, a connection can only be established from an VNC client (TCU, Ultr@VNC, etc.) by specifying the correct password.

Command `sc vncpwd set` always asks for a password to be entered.

Or, for example: sc vncpwd set systemnetwork secreT7+

Command `sc vncpwd reset` deletes the password for the specified network. A connection in the network is therefore possible without authentication.

**Note**

**Password in the network**

In the network (factory network or company network), in order to access any server (NCU) in the network from a TCU (client) via a VNC connection, all servers should use the same password.

# Service and diagnostics 3

## 3.1 Creating a service system

### Overview

To **backup** user data or the complete CompactFlash Card, proceed as follows:

- Creating a service system.
- Backing up the license
- Backup: Data backup on USB storage medium or on a network drive.
- System diagnostics
  - Output network settings
  - Changing the network settings
  - Starting the VNC Viewer

To **restore** the user data or the complete CompactFlash Card, proceed as follows:

- Restore/Recover: Restore the data from USB storage medium or from network drive.
- Carry out software update.
- Carry out firmware update.

### 3.1.1 How do you create a service system for the NCU?

### Purpose

In case service is needed, create a portable "Emergency Boot System" (EBS) on a USB memory. Thus you can start the booting of the NCU from the service system in order to carry out various service tasks, such as data backup or updates, in a service shell.

Two partitions are created on the service system:

- A Linux partition that is not displayed under Windows.
- An FAT partition, to backup files or save software updates.
  The FAT partition can be read and written to under Linux as well as from a Windows system.
  The FAT partition can be addressed under the path/data in a command shell under Linux.

## Scope of delivery

To create a service system on a USB storage medium, the following files are included on CD:

- an executable file installdisk.exe
- an image file for USB-FlashDrive
- a file with the latest information siemensd.txt / siemense.txt

**Recommendation:**

It is better to use SIMATIC PC USB-FlashDrive.

**Note**

To create the service system, you need administrator rights.

All data already on the USB storage medium will be deleted.

The transfer is optimized for USB 2.0; therefore, the transmission to the USB storage medium takes longer when using USB 1.1 than USB 2.0

## Proceed as follows

To create a service system on a USB storage medium:

1. Copy the service system onto a local hard disk of your PG/PC.
2. Connect the USB storage medium to the USB interface of the PG/PC.
3. Determine in Windows Explorer which drive letter the USB storage medium was assigned, e.g. H:
4. Open a DOS shell and change to the directory in which the files for the service system are stored.
5. In the DOS shell, enter the following command:
   ```
   installdisk --verbose --blocksize 1m linuxbase.img h:
   ```

**Result:** The image is transferred to the USB storage medium.

## 3.1.2 This is how you operate the service system

### Connecting the service system

Procedure:

1. Connect the USB storage medium to the USB interface X125 or X135 of the NCU.
2. Switch-off the system and then switch on again.
   OR
3. Press the "Reset" button.
   The service system is unzipped from the USB storage medium when the NCU boots for the first time: Only then can the USB storage medium be used as service system and the complete storage capacity is available.

### Operating the service system

Keys and softkeys to navigate in the service system:

| Softkey | Key on the OP | External key-board | Description |
|---|---|---|---|
| ↓ | HSK1 | <F1> | Moves the cursor down a row |
| ↑ | HSK2 | <F2> | Moves the cursor up a row |
| Page↓ | HSK3 | <F3> | Moves the cursor down a page |
| Page↑ | HSK4 | <F4> | Moves the cursor up a page |
| Char↓ | HSK5 | <F5> | Inserts text or digits |
| Char↑ | HSK6 | <F6> | Inserts text or digits |
| Cancel | VSK7 | <Shift> + <F7> | Cancel / Return |
| Ok | VSK8 | <Shift> + <F8> | OK / Confirm |
| --- | NEXT WINDOW | Pos1 | Moves the cursor to the top row |
| --- | END | End | Moves the cursor to the bottom row |

**Note**

**Operation with touch panels**

The service system supports touch operation. The input fields are touch operated and the integrated keyboard automatically appears.

For example: SINUMERIK OP 019 black, SIMATIC HMI Industrial Thin Client

## 3.1.3 Diagnostics functions

### Functions of the service menu

The following functions for service and diagnostics are available on the service system:

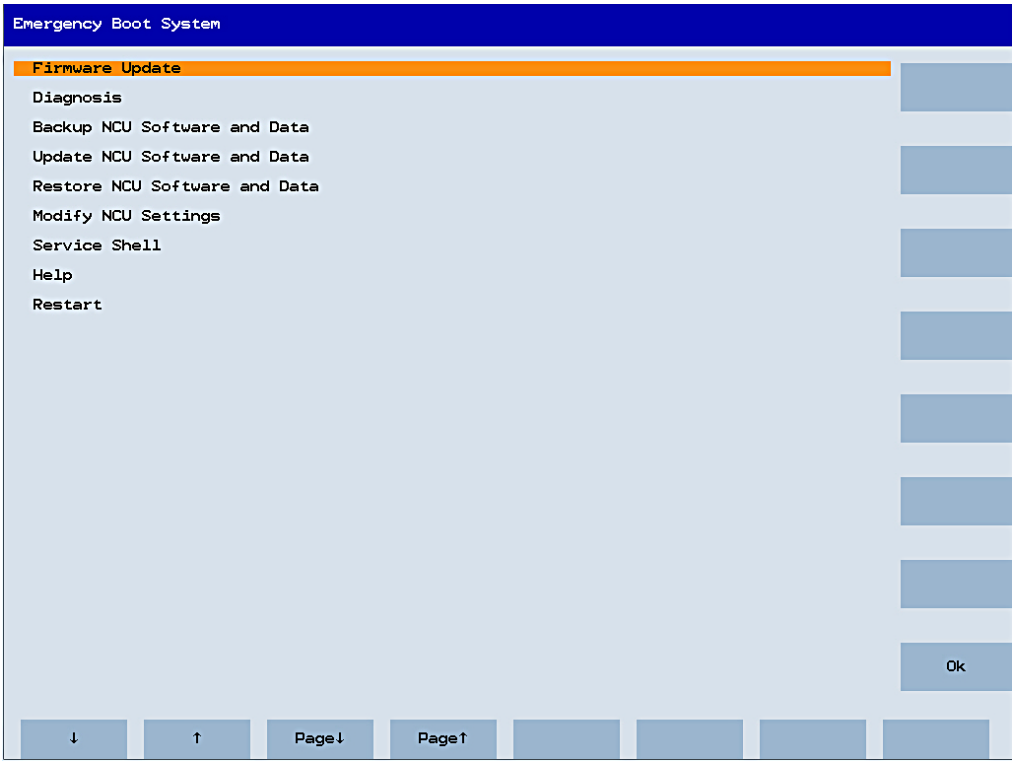



Figure 3-1 Service menu

Overview of the main menu:

| Main menu | Description | Reference |
|---|---|---|
| Firmware Update | Update BIOS:<br>• PLC BIOS<br>• FPGA<br>• CBE30 Bootloader | This is how you update the firmware (Page 84) |
| VNC Viewer | Starting the VNC viewer | Calling the VNC Viewer (Page 94) |
| Diagnosis | Starting a diagnostic function to display network-connection data and data on the system Compact-Flash Card. | Performing system diagnostics (Page 89) |
| Backup NCU Software and Data | Backup system and user data on the USB FlashDrive or network drive. | Backup data via the service menu (Page 72) |
| Update NCU Software and Data | Upgrade the system from the USB FlashDrive or network drive. | This is how you update the software (Page 86) |

| Main menu | Description | Reference |
|---|---|---|
| Restore NCU Software and Data | Restore system and user data from the USB storage medium or network drive. | Backup data via the service menu (Page 72) |
| Modify NCU Settings | • Change network settings<br>• Activate/deactivate HMI<br>• Restore delivery condition of the following components:<br>– HMI (SINUMERIK Operate)<br>– NC<br>– PLC<br>– Drive | This is how you change the system settings (Page 91) |
| Service Shell | A Service Shell is opened in order to input service commands. | Backing up data via the service shell (Page 80)<br>Service Commands (Page 38) |
| Help | Help text to operate the service menu in case the external keyboard has no function keys: | |
| | ALT 1 ... 8 | Function keys <F1> ... <F8> for operating the horizontal softkeys |
| | SHIFT ALT 1 ... 8 | Key combination for operating the vertical softkeys: VSK1 ... VSK8 |
| Restart | Restarts the system | |

## 3.1.4 This is how you connect to a network drive

### Connecting network drives

1. Select "Connect to network drive".
2. Enter the complete path name of the network drive.

3. Register yourself with a user name with access authorization to this drive.
   Note the syntax of the examples specified:

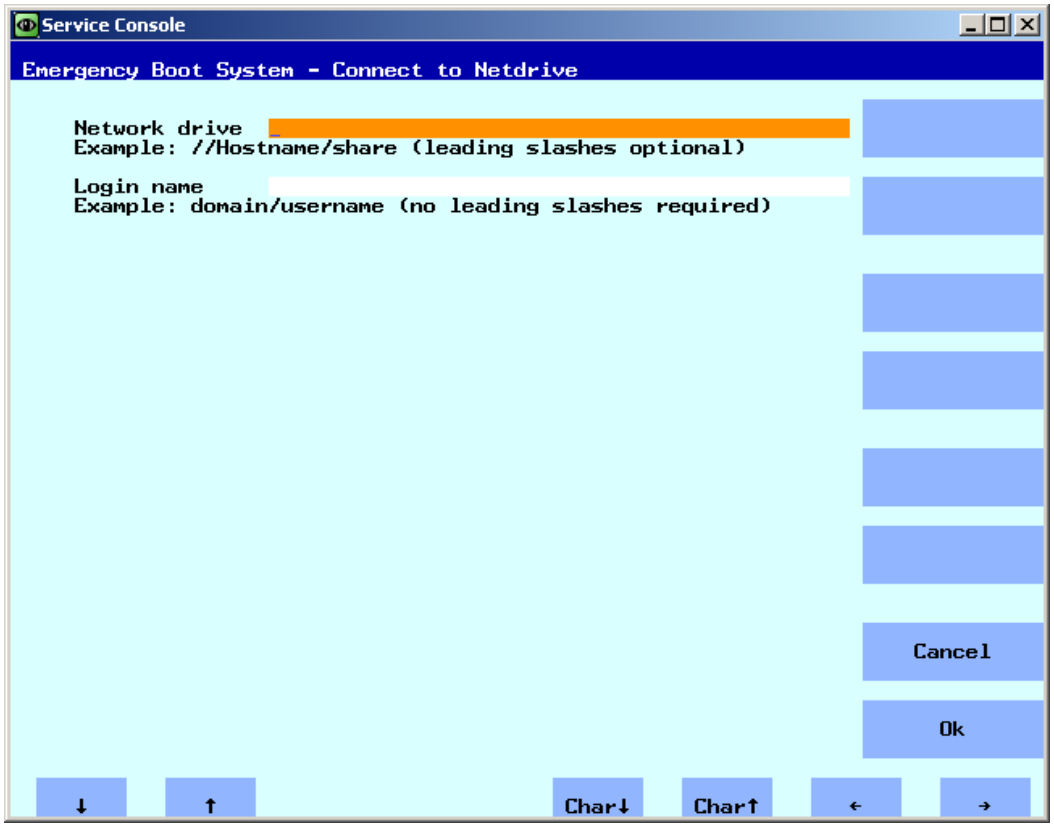



Figure 3-2 Connecting network drives

4. Confirm with "Ok". You are then prompted to enter a password.

# 3.2 Backup license

## Validity of the license key

The license key corresponds to the serial number of the CompactFlash Card. If the CompactFlash Card is replaced for a SINUMERIK 840D sl, the license key loses its validity and the system is no longer ready for operation.

This situation can occur for a defective hardware of the system CompactFlash Card.

## Application

For the following cases, backup the license key of the CompactFlash Card on a USB storage medium or on a network drive:

- Software update without license key
- System software error without defective hardware

**Note**

**Replacement CompactFlash Card**

The following data are required in order to obtain a valid license key after replacing the system CompactFlash Card: **The serial number of the defective and the new** CompactFlash Card.

## 3.2.1 This is how you backup the license key

### Backing up the license key on a USB storage medium

Procedure:

1. In the main menu, select the menu item "Backup NCU Software and Data".
   The following menu is displayed:

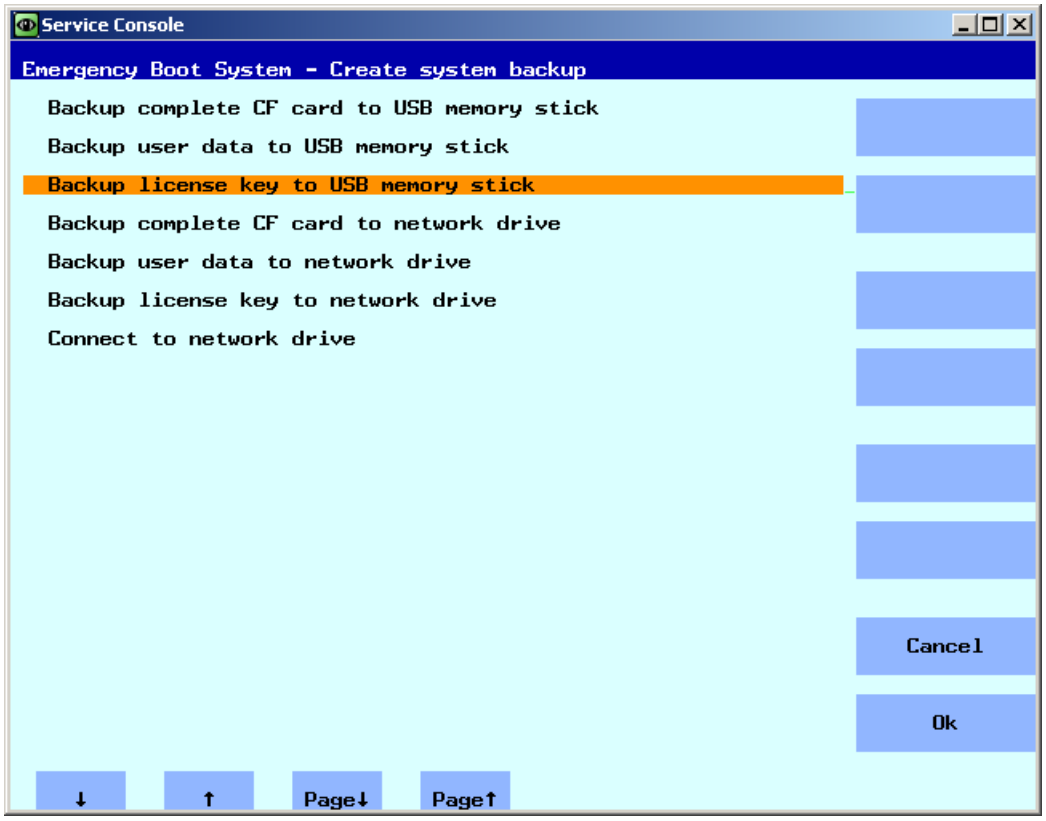



Figure 3-3 Backing up licenses

2. Select the menu item "Backup license key to USB memory stick".
   The following menu is displayed:

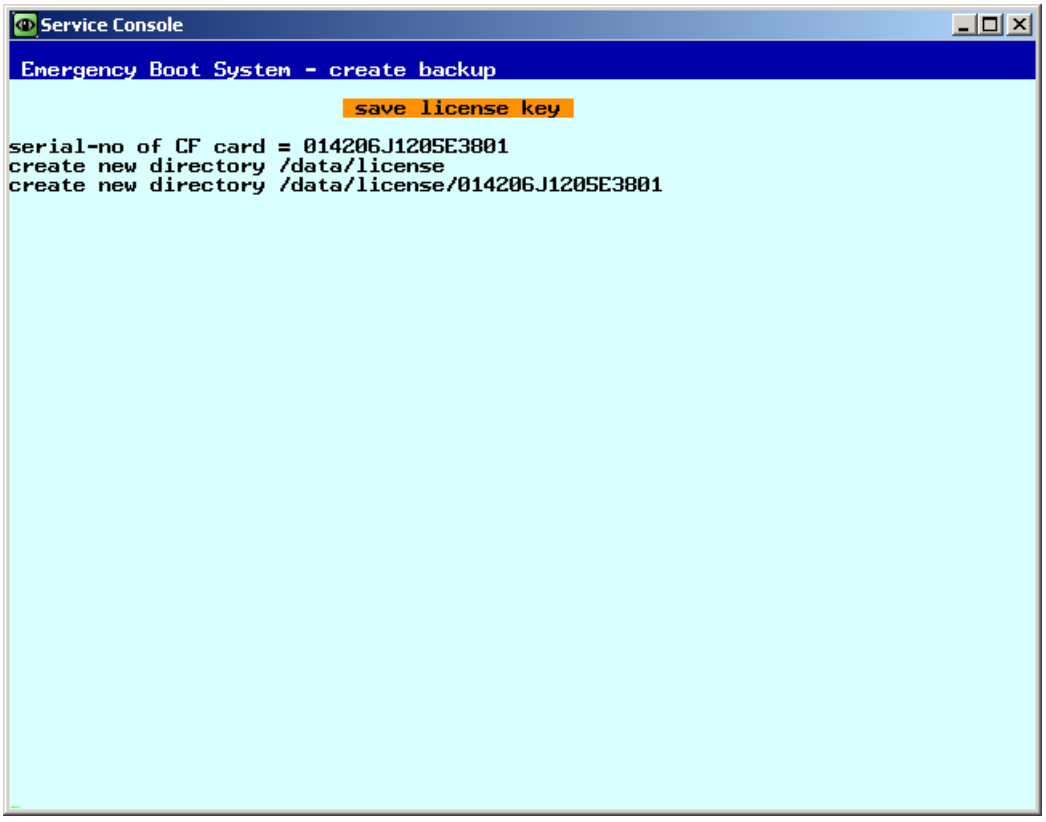



Figure 3-4 Save license key

The license key is backed up in the specified directory together with the serial number of the CompactFlash Card.

## See also

The procedure is essentially the same as for "Backup license key to network drive". In addition, you connect to a network drive:

This is how you connect to a network drive (Page 67)

# 3.3 Backup data via the service menu

## 3.3.1 This is how you backup user data

### Backing up and restoring user data

Procedure:

1. In the main menu, select the menu item "Backup NCU Software and Data".
2. Select "Backup user data to USB memory stick" and confirm with "Ok".

3. Enter a complete file name - as shown in the example:

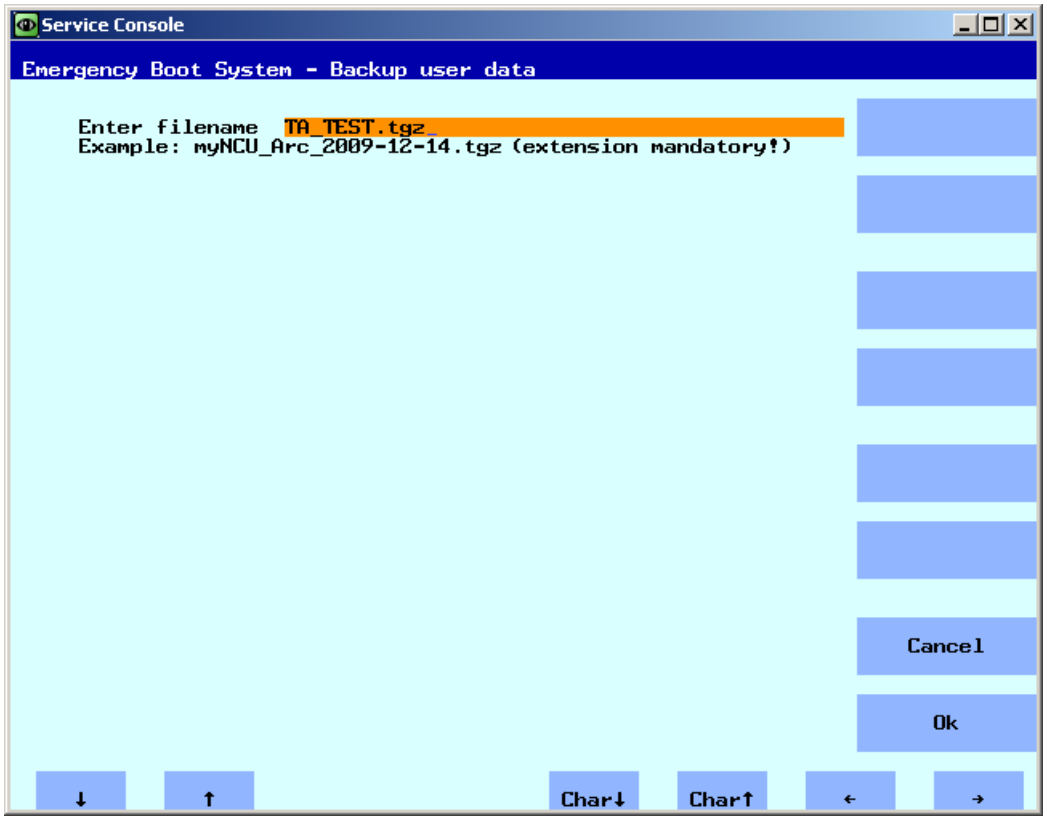



Figure 3-5 Entering a file name

4. Confirm with "Ok". The following message is output after the operation has been successfully completed:

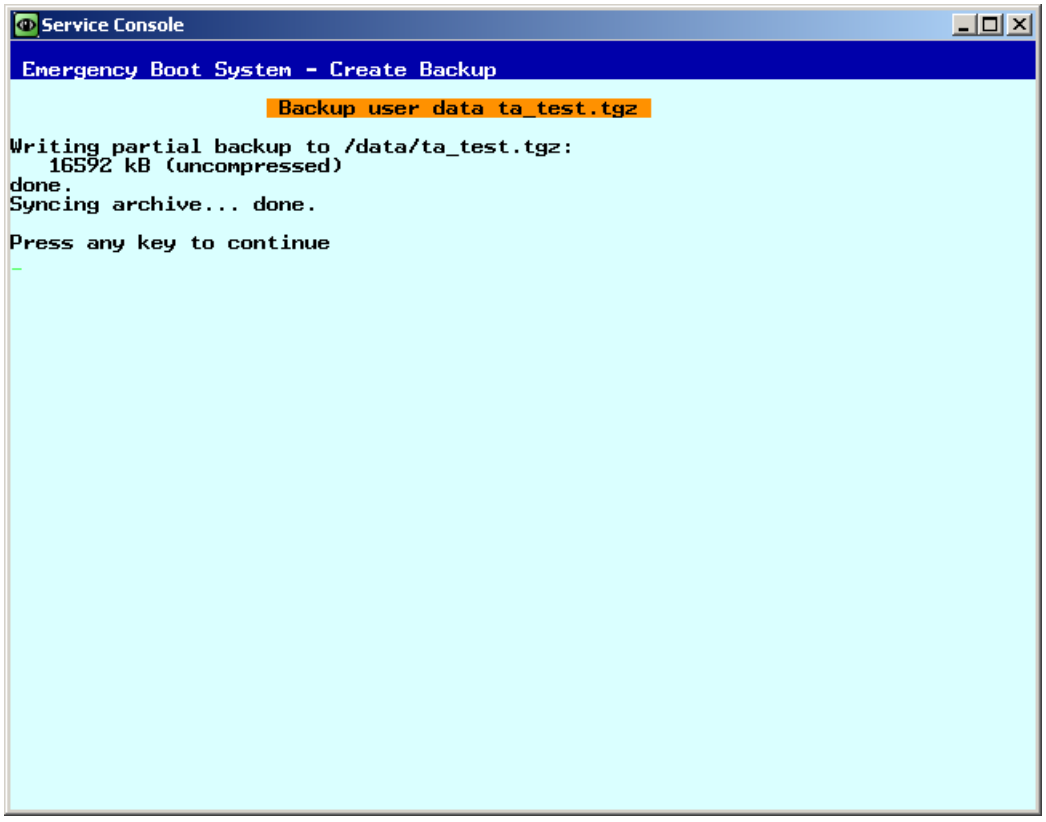



Figure 3-6 Operation completed

**Note**

**Backup file ≥ 4GB**

For a backup file ≥ 4GB, the space on the FAT partition of the service system (with 8GB) is no longer sufficient. For this reason, backup files of this size are partitioned (see also: sc save/sc restore) or they can be saved on a network drive.

## See also

The procedure for "Backup user data to network drive" is analog. In addition, you connect to a network drive:

This is how you connect to a network drive (Page 67)

## 3.3.2 This is how you restore user data

### Restoring user data

Procedure:

1. In the main menu, select the menu item "Restore NCU Software and Data".
   The following menu is displayed:

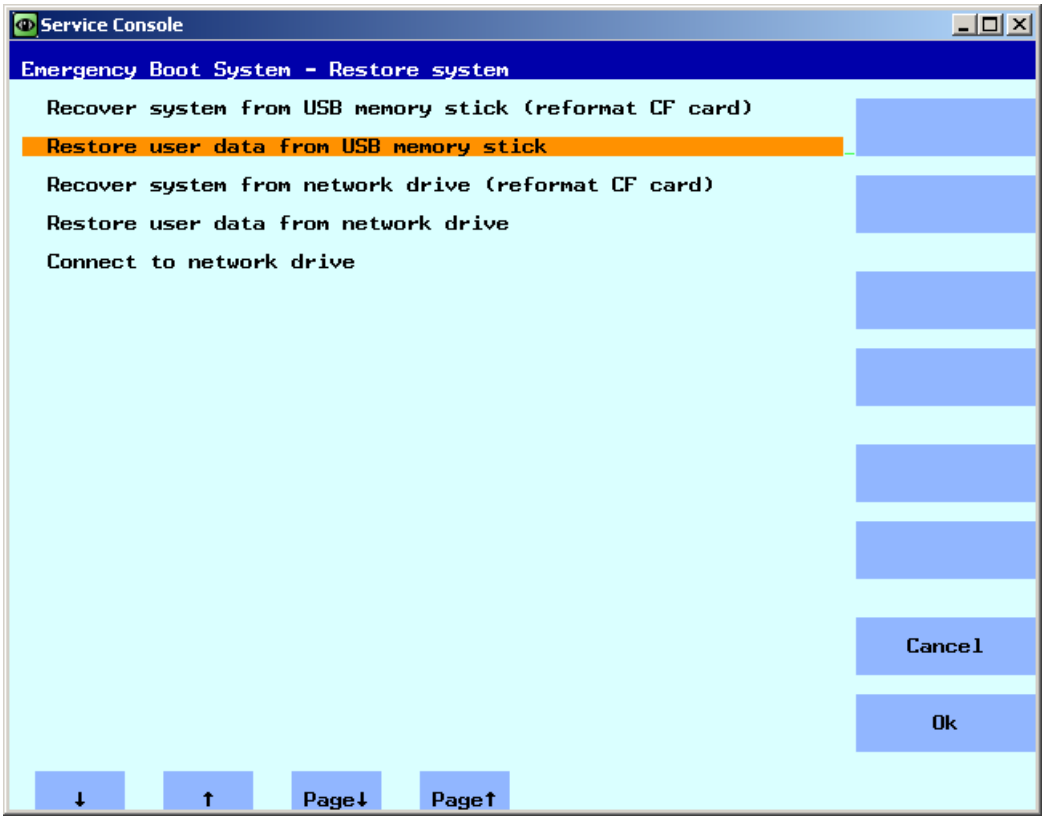



Figure 3-7 Restoring user data

2. Select "Restore user data from USB memory stick" and confirm with "Ok".
   The list of the tgz files available on the USB storage medium are displayed.
3. Select the appropriate tgz file and confirm with "Ok".
   A message is output after the operation has been successfully completed.

### See also

The procedure for "Restore user data from network drive" is essentially the same. In addition, you connect to a network drive:

This is how you connect to a network drive (Page 67)

### 3.3.3 This is how you create a complete backup of the CompactFlash Card

#### Creating a complete backup

Procedure:

1. In the main menu, select the menu item "Backup NCU Software and Data".
   The following menu is displayed:

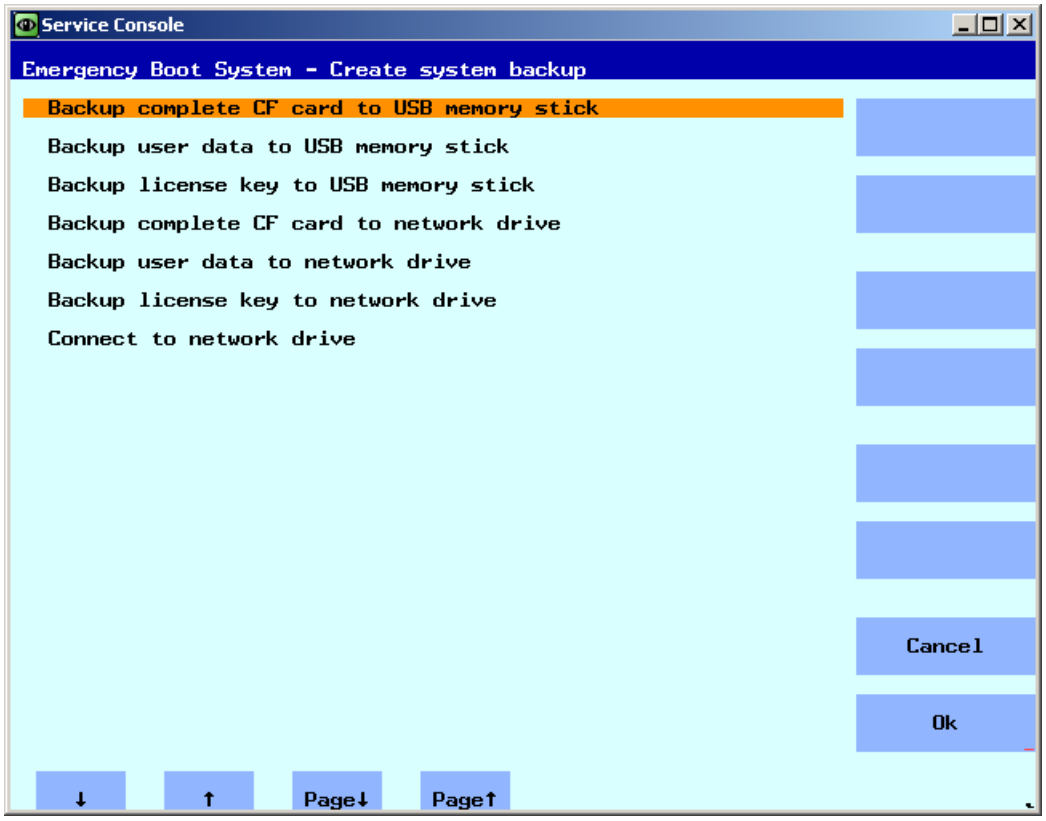



Figure 3-8 Backup up

2. Select "Backup complete CF card to USB memory stick" and confirm with "OK".

3. Enter a complete file name - as specified in the example.
4. Confirm with "OK". This can take several minutes.
   The operation is displayed:

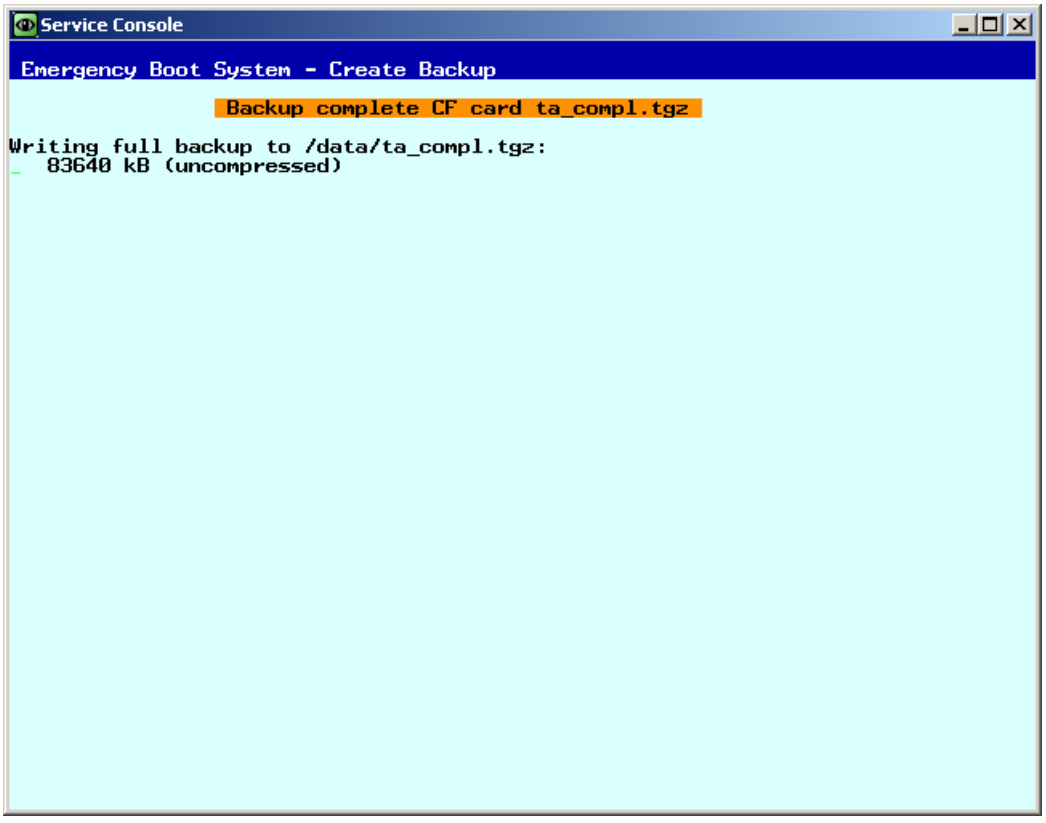



Figure 3-9 Backing up ...

A message is output after the operation has been successfully completed.

## See also

The procedure for "Backup complete CF card to network drive" is essentially the same. In addition, you connect to a network drive:

This is how you connect to a network drive (Page 67)

### 3.3.4 This is how you install a complete system backup

#### Installing the complete backup

With this menu item, a complete backup of the data from the service system or from a network drive is installed on the CompactFlash Card, i.e. this backup includes both the system software as well as also the user data.

| NOTICE |
| --- |
| **Formatting**<br>This procedure corresponds to formatting the CompactFlash Card and all of the existing data are overwritten. |

## Installing the complete backup

Procedure:

1. In the main menu, select the menu item "Restore NCU Software and Data".
   The following menu is displayed:

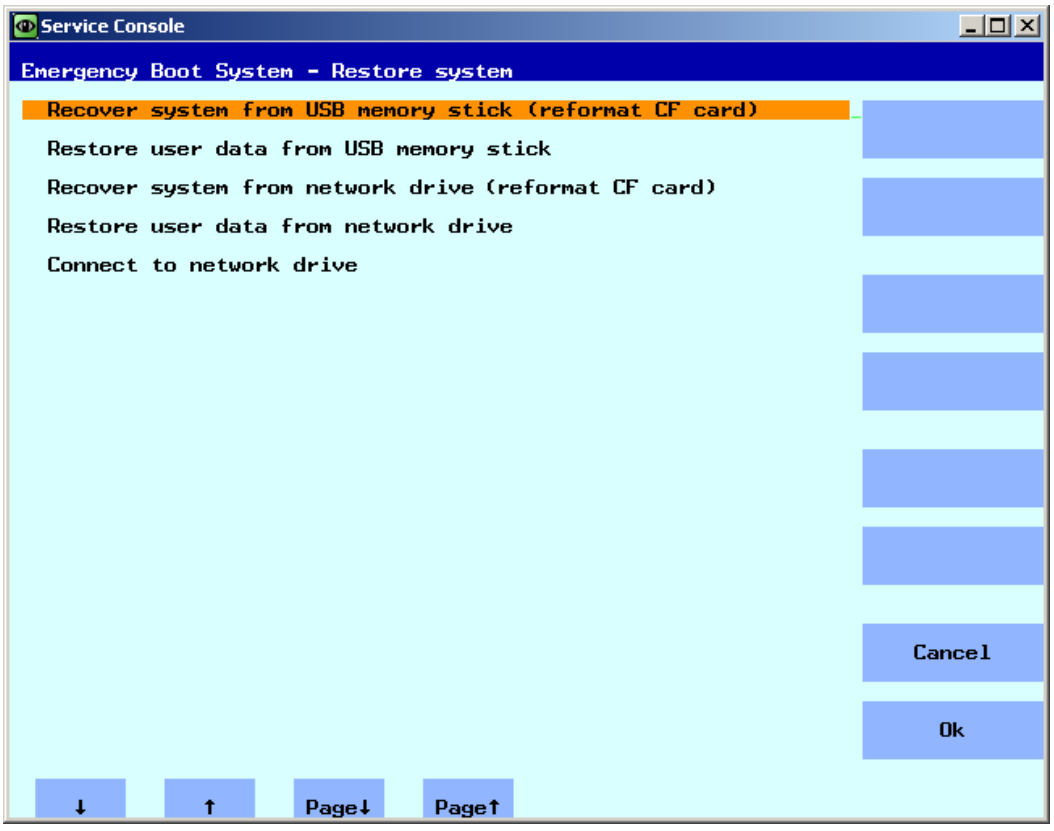



Figure 3-10 Restore system

2. Select the menu item "Recover system from USB memory stick (reformat CF card)" and confirm with "Ok".
   The list of the tgz files available on the USB storage medium are displayed.
3. Select the appropriate tgz file and confirm with "Ok".
   A message is output after the operation has been successfully completed.

## Completion

The system must be restarted in order that the new data become effective.

## See also

The procedure for "Recover system from network drive (reformat CF card)" is essentially the same. In addition, you connect to a network drive:

This is how you connect to a network drive (Page 67)

# 3.4 Backing up data via the service shell

### Backing up and restoring using a service shell

To call a service shell, depending on the configuration, the following possibilities are available:

(I) Configuration of NCU with TCU: the service shell is called under Linux.

(II) Configuring the NCU with PCU 50 or programming device (PG).

A service shell can be called under:

- on the NCU under Linux
- on the PCU/PG using the VNC Viewer (system network or company network)
- on the PCU/PG using WinSCP (system network or company network)

## 3.4.1 This is how you save data on a service system

### Proceed as follows

To back up the complete system:

1. Connect the service system to a USB interface (X125 or X135) of the NCU and press the reset button.
   Alternatively, you can switch the NCU off, connect the service system, and switch the NCU on again.
   **Result:** The NCU boots from the service system.
2. In the main menu, select the menu item "Service Shell".
3. Log on as a service technician using the user name "manufact" and password "SUNRISE".
4. Using the command "sc backup" you can create the backup file "backup01.tgz".
   The directory /data on the service system is provided for backup files. The complete path should be specified.
5. Choose –full, to backup all data on the CompactFlash Card, or –user, if you only want to backup user data.
   Example: `sc backup –full /data/backup01.tgz`
   **Result:**
   A backup file of the complete CompactFlash Card is created under /data on the service system on the USB storage medium.

## 3.4.2 This is how you restore data from the service system

### Proceed as follows

To restore the complete system:

1. Connect the service system to a USB interface (X125 or X135) of the NCU and press the reset button.
   Alternatively, you can switch the NCU off, connect the service system, and switch the NCU on again.
   **Result:** The NCU boots from the service system and the main menu is displayed.
2. In the main menu, select the menu item "Service Shell".
3. Log on as a service technician using the user name "manufact" and password "SUNRISE".
4. With the command "sc restore" you can write the backup file "backup01" from the service system back to the CompactFlash Card in the NCU. The complete path should be specified.
   Example: `sc restore /data/backup01.tgz`
   **Result:**
   The system state stored in the file "backup01" is restored on the NCU.

**Note**

If access to the system data on the CompactFlash Card is not possible because the CompactFlash Card is defective or empty, you can only log in as user "admin" with the password "SUNRISE" and no longer as the user "manufact".

## 3.4.3 This is how you save data on a network drive

### Sequence

Proceed as follows:

- Connect the service system.
- "Service Shell" open.
- Establish a connection to a network drive.
- Create a backup file.

### Scenario `I`: Start Command Shell under Linux

Procedure:

1. Connect the service system to the USB interface X125 or X135 of the NCU.
2. In the main menu, select the menu item "Service Shell".
3. Log on as a service technician using the user name "manufact" and password "SUNRISE".

4. Use the command "sc connect" to connect the network drive:
   `sc connect //username%password@server/share /tmp/backup`
   Please enter here the user name and the password for the network drive that is to be connected to.
5. Using the command "sc save", create the backup file "backup01".
   Choose –full, to backup all data on the CompactFlash Card, or –user, if you only want to backup user data in the directory /user.
   Example: `sc save –full /tmp/backup/backup01`
   **Result:**
   A backup file of the complete CompactFlash Card is created under the specified path on the network drive.

### Scenario II: Start the command shell with WinSCP on the programming device:

Procedure:

1. Start WinSCP and enter the following data in the log-on window:
   – IP address of the NCU (or, if required, host name)
   – User name "manufact" with password "SUNRISE".
2. Select in the menu "Commands" → "Open Terminal".
3. Execute the data backup using the commands described in steps 4 to 6 from scenario I.

### Scenario II: Start the command shell using the VNC Viewer on the programming device

Procedure:

1. Start the VNC Viewer and connect to the NCU using the IP address (or where relevant, using the host name)
2. Log on under the "manufact" user name with the password "SUNRISE".
3. To restore the data, enter the corresponding commands as described under steps 3 to 5 in scenario I.

## 3.4.4 This is how you restore data from the network drive

### Sequence

Proceed as follows:

- Connect the service system.
- "Service Shell" open.
- Establish a connection to a network drive.
- Restore the data.

### Scenario I: Start command shell under Linux

1. Connect the service system to the USB interface X125 or X135 of the NCU.
2. In the main menu, select the menu item "Service Shell".
3. Log on as a service technician using the user name "manufact" and password "SUNRISE".
4. Use the command "sc connect" to connect the network drive:
   `sc connect //username%password@server/share /tmp/backup`
5. Please enter here the user name and the password for the network drive that is to be connected to.
   To completely restore the system (system data and user data), enter the following: `sc restore –full backup01`
   **Result:** The whole system is overwritten by the backup data.
6. To restore user data only, enter the following command:
   `sc restore –user backup01`
   **Result:** The user data are written back again.
   All subsystems are then restarted: `sc start all`
   **Result:**
   The system state stored in the file "backup01.tgz" is restored on the NCU.

### Scenario II: Start the command shell with WinSCP on the programming device:

Procedure:

1. Start WinSCP and enter the following data in the log-on window:
   – IP address of the NCU (or, if required, host name)
   – User name "manufact" with password "SUNRISE".
2. Select in the menu "Commands" → "Open Terminal".
3. To restore the data, enter the corresponding commands as described under steps 3 to 5 in scenario I.

### Scenario II: Start the command shell using the VNC Viewer on the programming device

Procedure:

1. Start the VNC Viewer and connect to the NCU using the IP address (or where relevant, using the host name)
2. Log on under the "manufact" user name with the password "SUNRISE".
3. To restore the data, enter the corresponding commands as described under steps 3 to 5 in scenario I.

# 3.5 Updating the software

## 3.5.1 This is how you update the firmware

### Firmware Update

When service is required, it may be necessary for a service technician to install a BIOS update. Siemens provides the corresponding update file. Possible formats include: *.img or *.rom or *.bin. Or, the update file is already supplied on the service system, then step 1 is eliminated.

Procedure:

1. Copy the update file to the FAT partition of the service system.
2. Connect the service system to interface X125 or X135 of the NCU.
3. Switch on the system.
4. In the main menu, select the menu item "Firmware Update".
   The following menu is displayed:

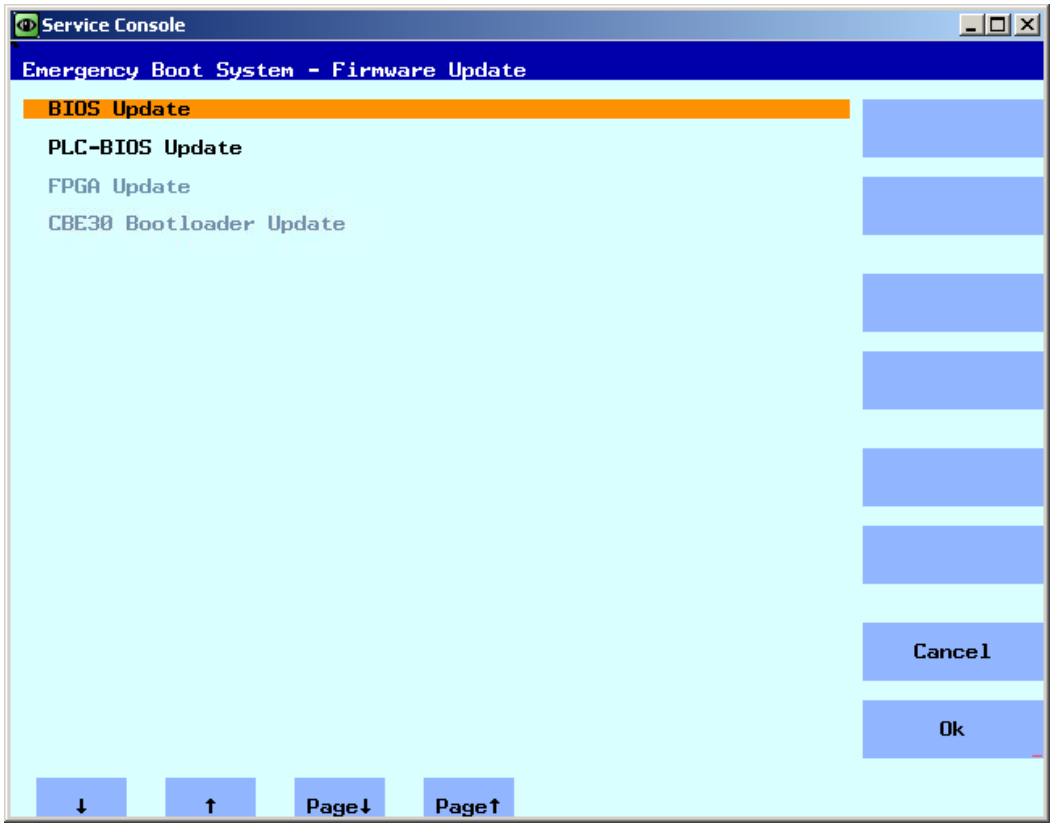



Figure 3-11 Firmware update

5. Select "BIOS Update" and follow the next instructions.
   The firmware version installed on the NCU and the update files on the service system are output:

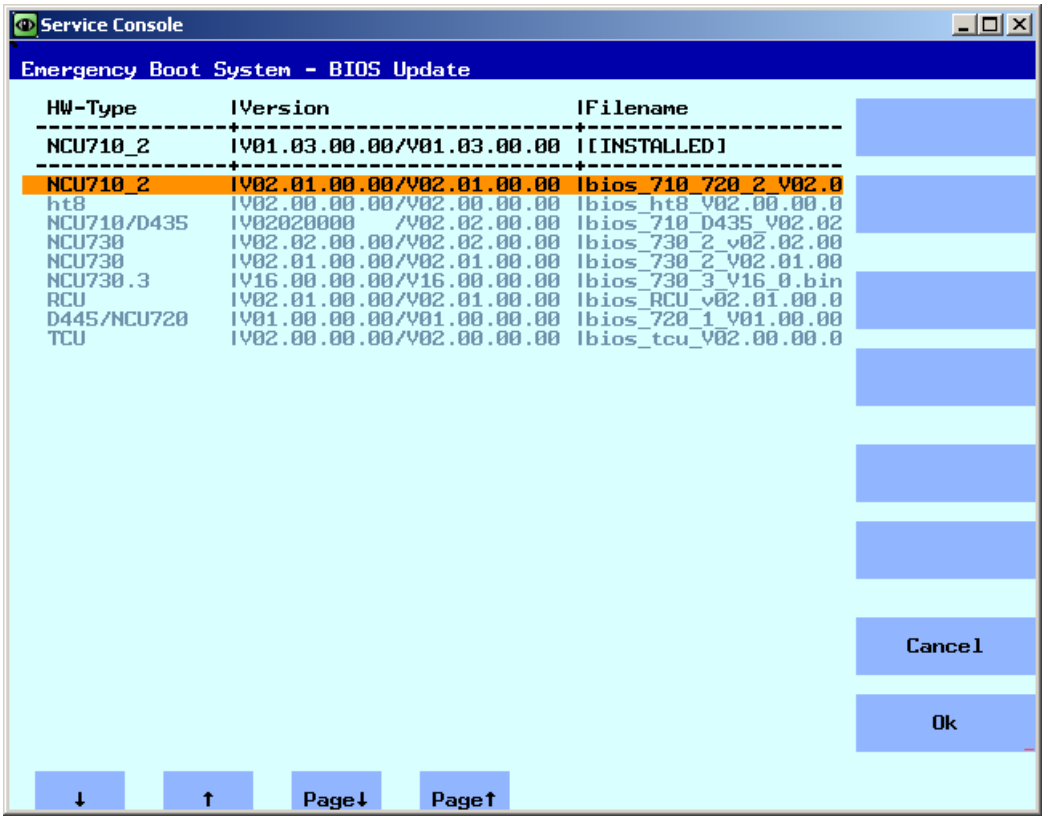



Figure 3-12 Select the updates

6. Select the update file and confirm with "OK".
   A "Restart" is then required.

**Note**

**Fall back strategy**

Before the operating system update is installed, a backup of the installed firmware version is stored on the service system.

For safety reasons, only files can be selected that are suitable and are appropriate for this NCU.

If a directory called "BIOS" is created on the service system on the FAT partition, the operating system updates are found in this directory and the backup of the installed firmware version is stored in this directory.

### Update other components

The following rules must be observed for the additional updates:

- The procedure for updating the PLC-BIOS Update is the same as for the operating system update.
- FPGA Update: for NCU7x0.3
- CBE30 Bootloader Update: for CBE30 and CBE30-2 (option module)

## 3.5.2 This is how you update the software

### Update / Recover

The following options are available:

- Update the software from the service system or from the network drive.
- Completely restore the CompactFlash Card with the system software from the service system or from the network drive (Recover system ...).

## Updating the software

Procedure:

1. In the main menu, select the menu item "Update NCU Software and Data".
   The following menu is displayed:

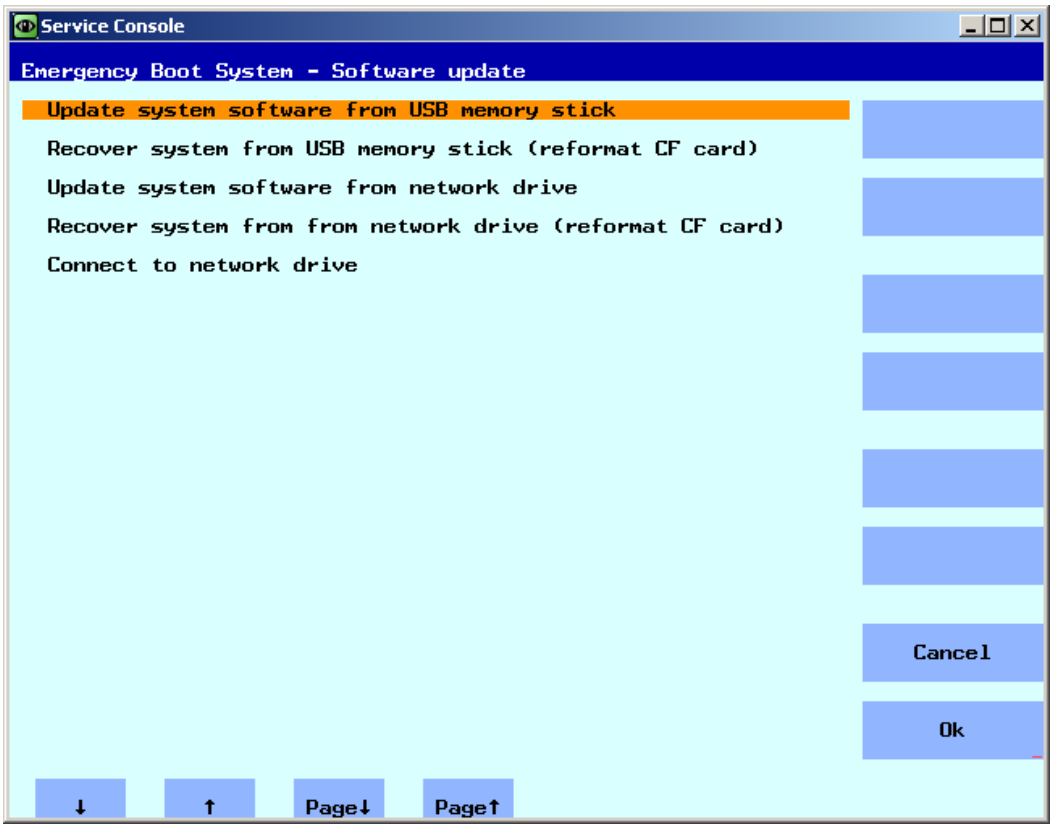



Figure 3-13 Software update

2. Select the menu item "Update system software from USB memory stick".
   The list of the tgz files available on the USB storage medium are displayed.

3. In order to check which software version contains the tgz file, first select the tgz file, then press "Show version".
   The following information is output:

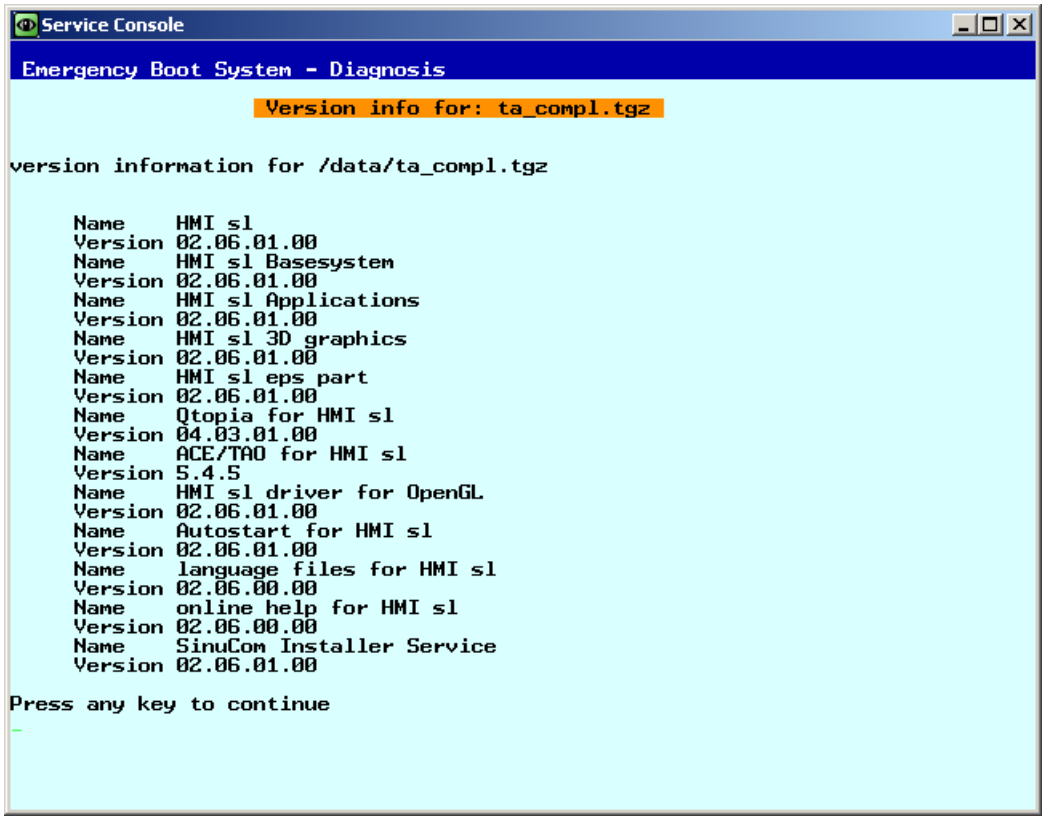



Figure 3-14 Version information

4. Select the appropriate tgz file and confirm with "Ok".
   A message is output after the update has been successfully completed.

## Completion

The system must be restarted in order that the new data become effective.

## See also

The procedure for "Update system software from network drive" is essentially the same. In addition, you connect to a network drive: This is how you connect to a network drive (Page 67)

For a complete backup of the system: This is how you install a complete system backup (Page 78)

# 3.6 System diagnostics

## 3.6.1 Performing system diagnostics

### Overview of diagnostic functions

The following diagnostic functions to display network settings are included in this menu:

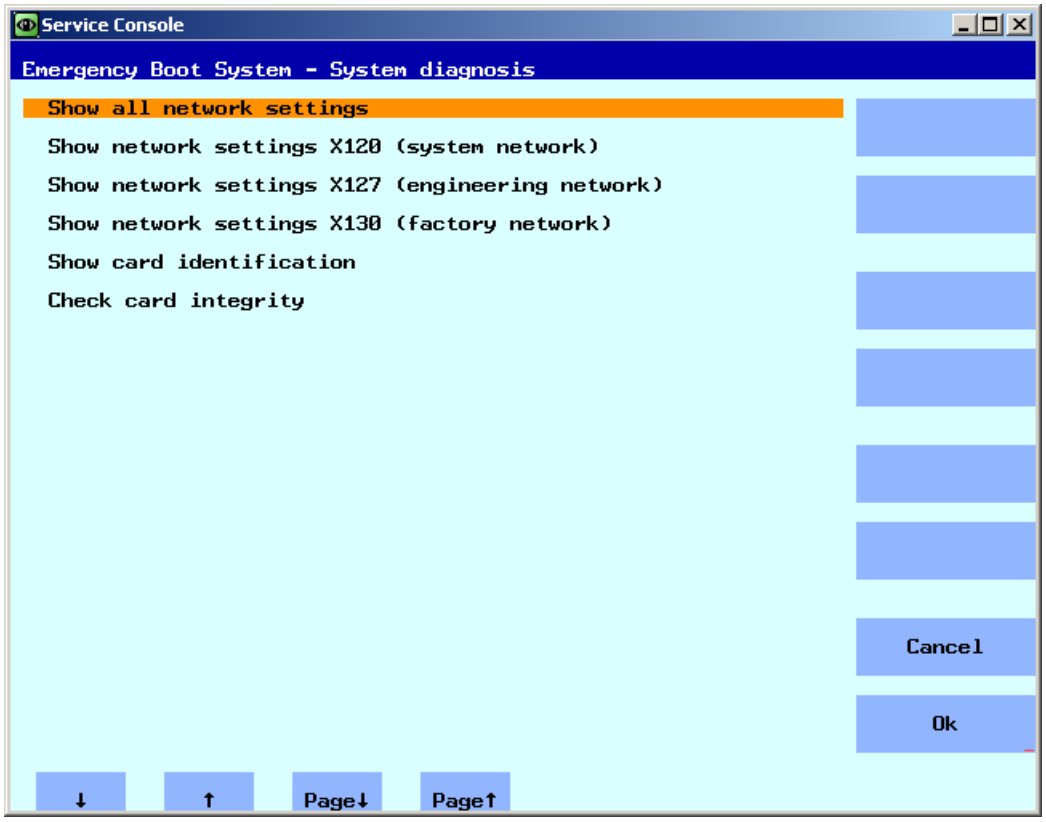



Figure 3-15 Diagnostics functions

| System diagnosis | Description |
|---|---|
| Show all network settings | Connection data of all Ethernet interfaces is displayed. |
| Show network settings X120 (system network) | The connection data of interface X120 is displayed. |
| Show network settings X127 (engineering network) | The connection data of interface X127 is displayed. |
| Show network settings X130 (factory network) | The connection data of interface X130 is displayed. |

| System diagnosis | Description |
|---|---|
| Show card identification | The following data of the system CompactFlash Card is output:<br>• Serial number<br>• System software version<br>• Hardware version (NCU) |
| Check card integrity | Checking the CompactFlash Card system |

**Example:** Connection data of all Ethernet interfaces of the NCU:

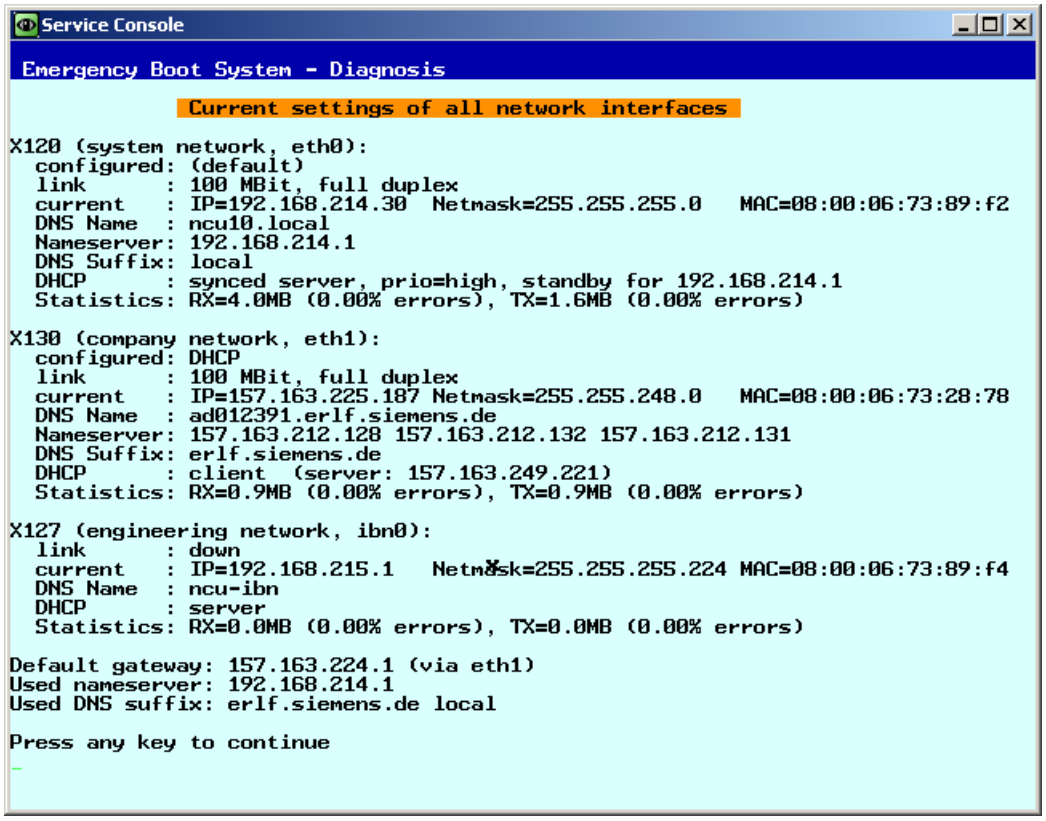



Figure 3-16 NCU: Interfaces

## See also

This is how you check the CompactFlash Card (Page 92)

## 3.6.2 This is how you change the system settings

### Modify NCU Settings

You can change the following system settings in this menu:

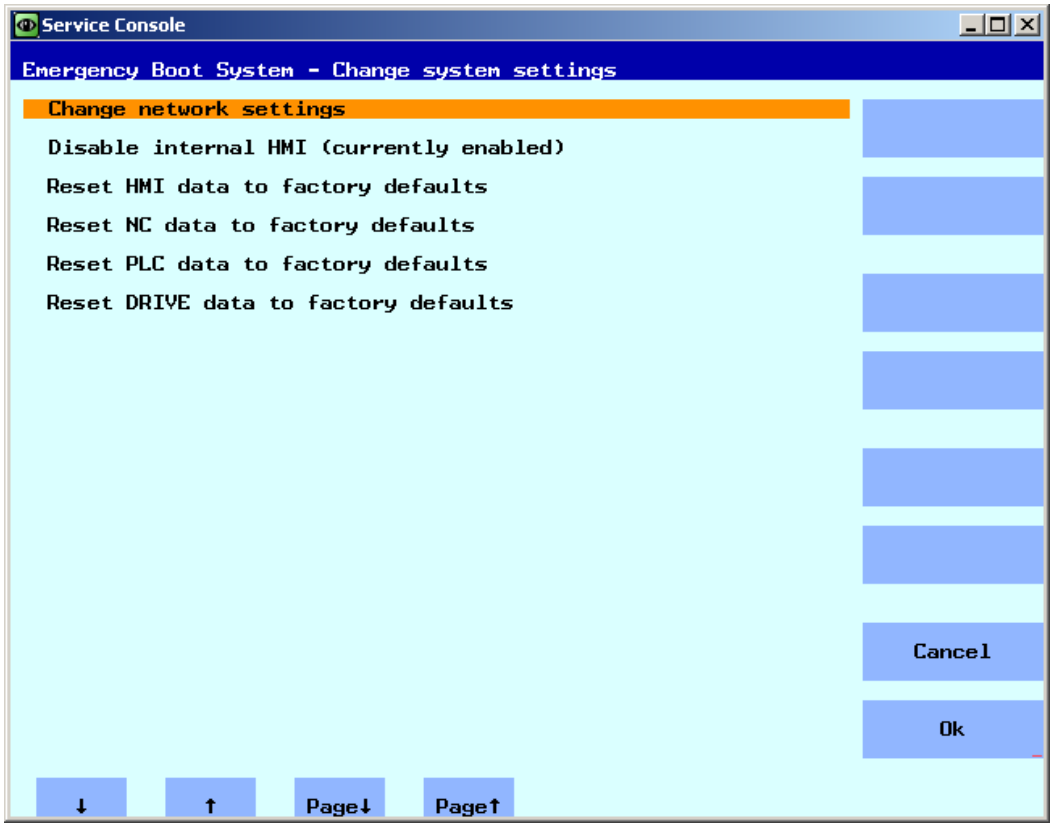



Figure 3-17 Changing system settings

| Menu item | Description |
|---|---|
| Change network settings | Changing network settings of the NCU. |
| Disable internal HMI (currently enabled) | Activating/deactivating an HMI task |
| Reset HMI data to factory defaults | Restoring the delivery condition of HMI data. |
| Reset NC data to factory defaults | Restoring the delivery condition of NC data. |
| Reset PLC data to factory defaults | Restoring the delivery condition of PLC data. |
| Reset DRIVE data to factory defaults | Restoring the delivery condition of drive data. |

| NOTICE |
|---|
| **Reset ... data to factory defaults**<br>The selected data are reset to the state when delivered, therefore overwriting all of the previous specific commissioning settings that have been made. |

## 3.6.3 This is how you check the CompactFlash Card

### Displaying identification data

Procedure:

1. In the main menu, select the menu item "Diagnosis".
   The following menu is displayed:

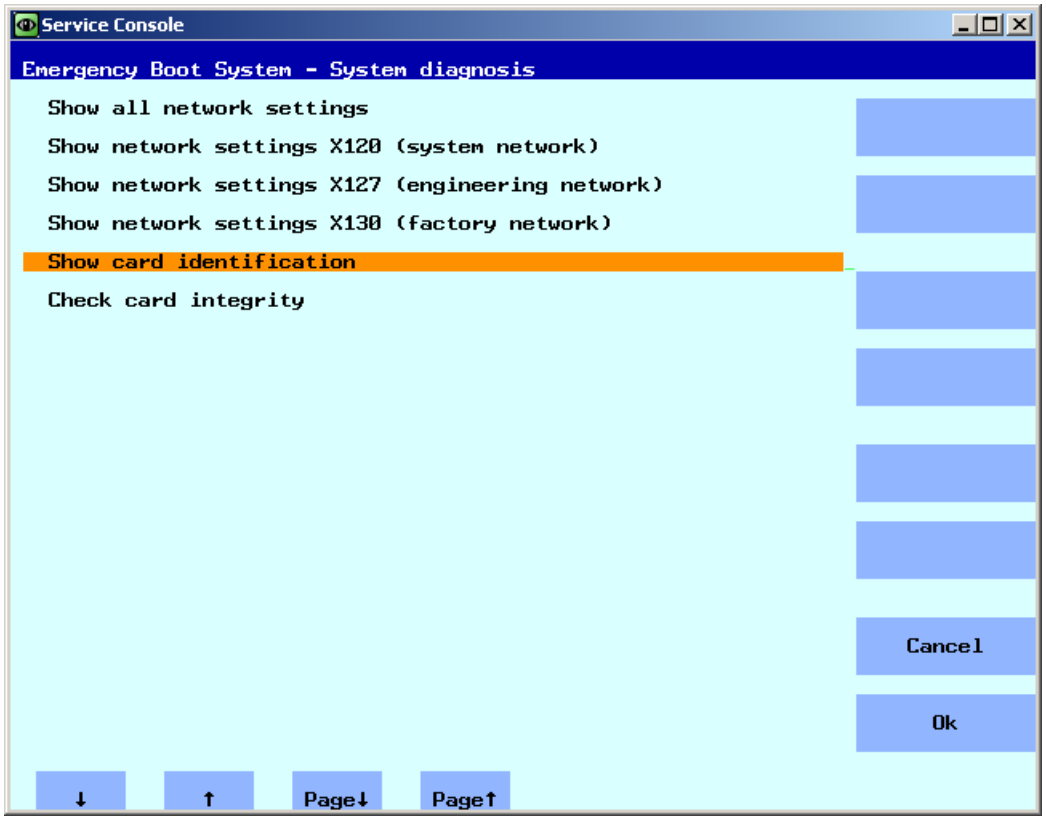



Figure 3-18 System diagnostics

2. Select the menu item "Show card identification".
   The following data for identification are output:

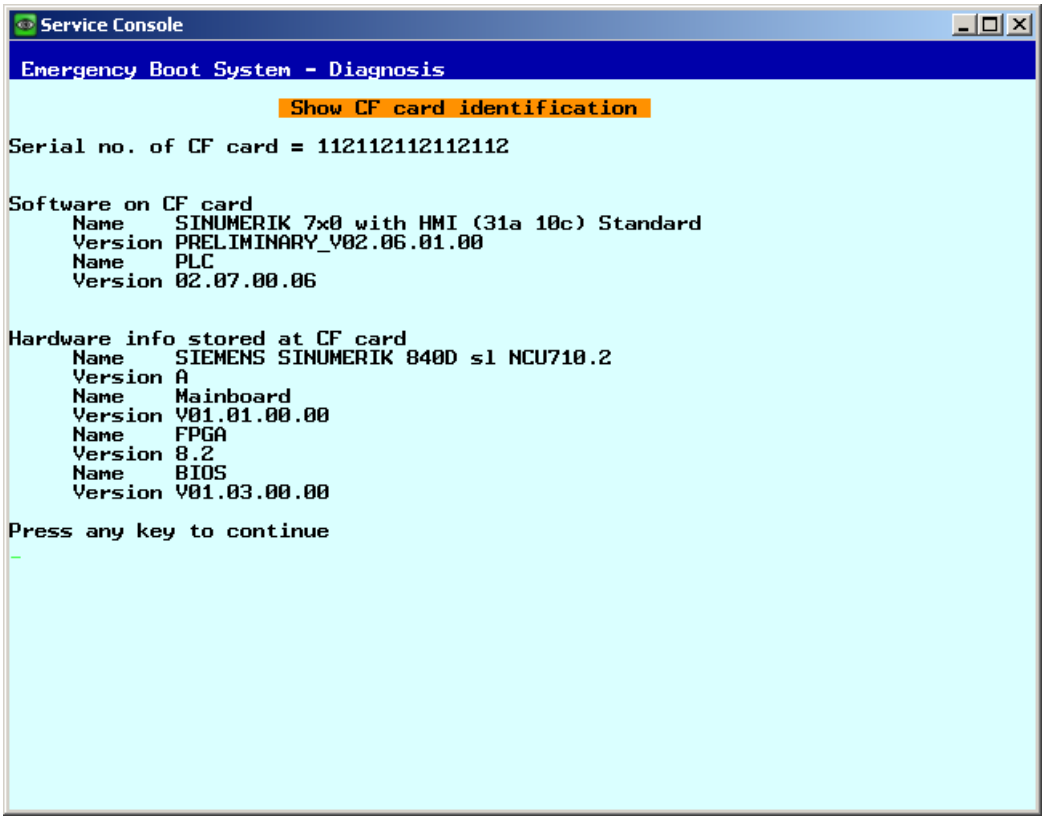



Figure 3-19 Identification

3. Select the menu item "Check card integritiy".
   The CompactFlash Card is checked for errors:

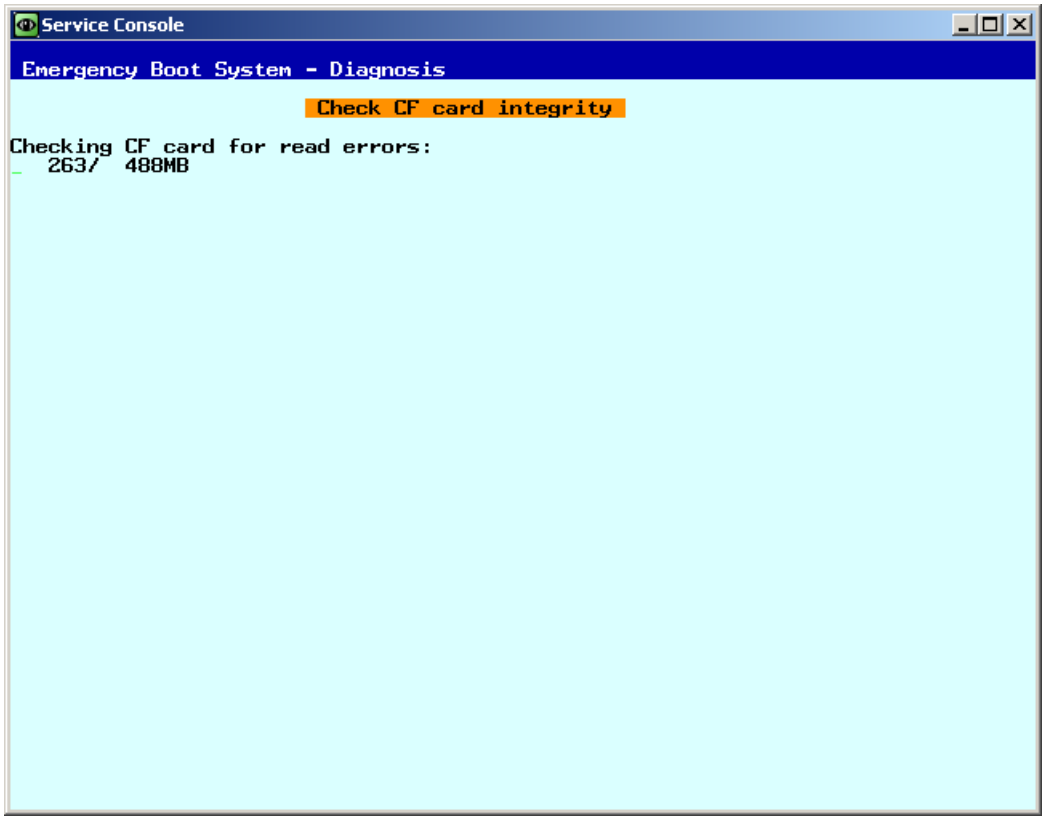



Figure 3-20 Check

**Note**

This procedure can take several minutes.

## 3.6.4 Calling the VNC Viewer

### Application

The VNC Viewer is used to connect directly with a VNC Server and, for example, to operate an HMI application.

Furthermore, it enables you to call up a list of all the network devices, thereby obtaining an overview of the system network.

## Starting the VNC Viewer

Procedure:

1. Select the "VNC Viewer" menu item in the main menu.
   The following menu is displayed:

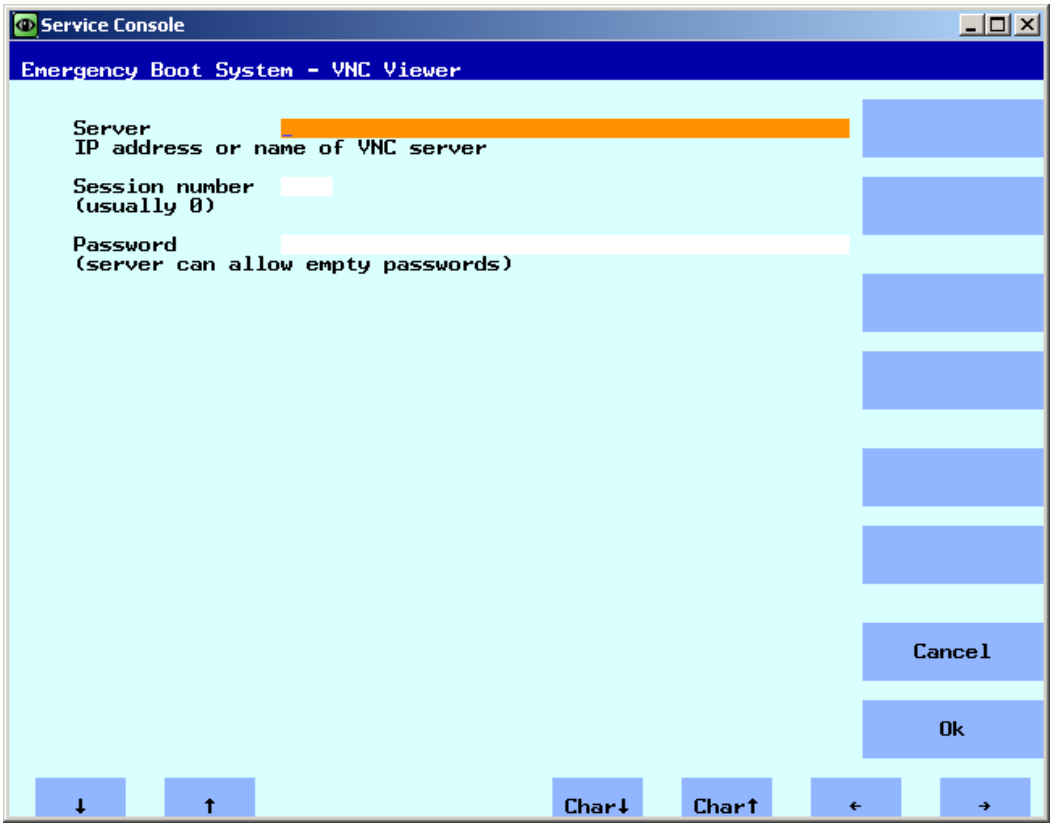



Figure 3-21 VNC Viewer

2. To start the VNC Viewer, enter the following connection data:
   - Server name, e.g. IP address or DNS.
   - The password is optional and can also be left empty.
   - The following assignment applies to the "Session number":

| Number | Meaning |
|---|---|
| Session 0 | HMI-Applikation |
| Session 4 | Command Shell |
| Session 5 | System logfile |

## 3.6.5 WinSCP and PuTTY

### Use

The programs WinSCP and PuTTY are freely-available open source programs for Windows. WinSCP is intended especially for transferring files from and to Linux systems, PuTTY for the remote operation of a command shell.

- WinSCP can be downloaded via the following link: http://winscp.net/eng/download.php (http://winscp.net/eng/download.php) (Installation Package).
  WinSCP also offers a "command shell" that is limited so that commands can be issued, but no callbacks can be answered.
- PuTTY, by contrast, offers a complete command shell.
  PuTTY web page: http://www.chiark.greenend.org.uk/~sgtatham/putty (http://www.chiark.greenend.org.uk/~sgtatham/putty)

With both programs, a service technician can log onto the NCU and carry out service tasks. The username ‘manufact’ with the password ‘SUNRISE’ is available for the Siemens service technician.

### Starting WinSCP

WinSCP is started from Windows after the service technician has logged onto the NCU with which he is connected using the corresponding authorization (e.g. as user "manufact" with password "SUNRISE").

From the "Commands“ menu, select ”Open terminal” to open a command shell. There you can carry out the service commands in the usual way.

# Appendix

# A

## A.1 Abbreviations

| | |
|---|---|
| CF | CompactFlash card: Memory card |
| CFS | Cluster File System |
| CIFS | Common Internet File System |
| DCK | Direct Control Keys: Direct control keys |
| DCP | Discovery and Basic Configuration Protocol |
| DHCP | Dynamic Host Configuration Protocol: Dynamic assignment of an IP address and other configuration parameters on a computer in a network |
| DNS | Domain Name System: Conversion of domain names into IP addresses |
| EBS | Emergency Boot System |
| EKS | Electronic Key System: System to check the identity of a user (authentication system) |
| EUNA | End User Notification Administration |
| HMI | Human Machine Interface: Operator interface |
| IRT | Isochronous Realtime (Ethernet) |
| LLDP | Link Layer Discovery Protocol: multi-vendor Layer 2 Protocol defined in accordance with the IEEE-802.1AB standard, allows information to be exchanged between devices. |
| MAC | Media Access Control: The MAC address is a 48-bit Ethernet ID. |
| MCP | Machine Control Panel: Machine control panel |
| MPI | Multi-Point Interface: Multiple interface |
| MUI | Multilanguage User Interface |
| NAT | Network Address Translation |
| NCK | Numerical Control Kernel: NC kernel with block preparation, travel range, etc. |
| NCU | Numerical Control Unit: NCK hardware unit |
| NFS | Network File Service is a network protocol. |
| NRT | Non-Realtime (Ethernet) |
| NTFS | New Technology File System |
| NTLMSSP | NT LAN Manager (NTLM) Security Support Provider |
| NTP | Network Time Protocol: Standard for synchronizing clocks in the entire network |
| NTPD | NTP Daemon:<br>Utility that runs in the background and does not have to be started by the user. |
| PCU | PC Unit: Computer unit |
| PDEV | Physical device |
| PG | Programming device |
| PLC | Programmable Logic Control: Programmable logic controller |
| RAM | Random Access Memory: Program memory which can be read and written into |
| RDY | Ready: The system is ready to operate. |
| RFC | Remote Function Call |
| SMB | Server Message Block |

| | |
|---|---|
| SNMP | Simple Network Management Protocol (network protocol for monitoring and controlling network elements such as routers, servers, switches, and printers from a central station). |
| SSD | Solid State Drive |
| SSH | Secure Shell: Protocol for an encrypted network connection with a remote device |
| TCU | Thin Client Unit |
| TFTP | Trivial File Transfer Protocol: Very simple data transmission protocol |
| UDP | User Datagram Protocol: NTP is mostly processed via UDP. |
| USB | Universal Serial Bus |
| UPS | Uninterruptible power supply |
| UTC | Universal Time, Coordinated: Coordinated Universal Time (previously: Greenwich Mean Time) |
| VNC | Virtual Network Computing |

# Glossary

### CFS (Compressed File System)

A CFS (file extension ".cfs") is a compressed file system, similar to a zip file. It contains files and subdirectories that look like normal files on the controller at runtime. Files and directories contained in a CFS cannot be changed. They are decompressed at runtime as required.

### Network interface

The network interface is an interface that enables network communication. These are the Ethernet interfaces on the NCU.

### NFS (Network File System)

NFS is the most common protocol for remote file systems in the world of Unix, and is also available for Windows. NFS is closely based on the Unix privilege model – each time a file is accessed, a UID and GID are supplied which the server then uses to decide whether the operation is permitted. The server relies on the client to provide the correct IDs.

### Remote File System

A file system that is contacted over the network. The files are physically located on another computer in the network (the "server"), but appear locally the same as all other files. Operations performed on these files are sent via the network to the server, instead of being executed directly on a local storage medium (such as a hard drive or CompactFlash Card).

As a server usually exports more than one file system, a name for the required file system must also be entered in addition to the name of the server.

### SMB (Server Message Block)

SMB is the underlying protocol of MS Windows file systems (also known as drives, releases, shares, etc.). SMB connections are always active in the context of a specific user, who must be known to the server. Exported file systems have a name (release name), by which they can be addressed. The client does not need to know the concrete path on the server.

### Subsystem

A subsystem is a CFS that not only contains a collection of files, but also executes a program, for example, at runtime. To do this, the CFS contains a script that is used to control the starting and stopping of this program.

For this reason, only administrators are permitted to set up NFS file systems, and NFS is usually only implemented in uniformly administrated environments. Exported file systems on the server are addressed directly on the server via their path.

## VNC (Virtual Network Computing)

Virtual Network Computing is a software that displays the screen contents of a remote computer, with a running VNC server, on a local computer, with a running VNC viewer, and in return sends keyboard and mouse movements of the local computer to the remote computer.

# Index


## 7

7-segment display, 10

## A

Applications, 63

## B

basesys.ini, 15
- CheckTCUforUpdatePackage, 30
- DefaultCIFSSecurity, 29
- DHCPClientID, 18
- DHCPDNoMasterWait, 20
- DisableCompanyNet, 16
- DisableIBNForwarding, 22
- DisableIBNNet, 22
- DisableNATRouting, 21
- DisableNTPTimeSync, 29
- DisablePLCTimeSync, 29
- DisableSubsystems, 28
- DisableSysNet, 18
- Domain, 18
- EnableCoreDumps, 29
- EnableDHCP_IBNNet, 21
- EnableDHCPD_SysNet, 19
- EnableSysNetToIBNForwarding, 22
- EnableUsbShares, 30
- ExternalDcpEnabled, 24
- ExternalIP, 16
- ExternalLldpEnabled, 24
- ExternalLldpTLVsTxEnabled, 25
- ExternalNetMask, 16
- FirewallOpenPort, 28
- Gateway, 16
- Hostname, 17
- InternalDcpEnabled, 24
- InternalDNSDomain, 21
- InternalDynRangeEnd, 20
- InternalDynRangeStart, 20
- InternalIP, 18
- InternalIP_Alias, 19
- InternalLldpEnabled, 24
- InternalLldpTLVsTxEnabled, 25
- InternalNetMask, 19
- InternalNetMask_Alias, 19
- LogFileLimit, 28
- Name servers, 17
- PLCTimeType, 27
- SNMPAutLocation, 23
- SNMPContact, 23
- SNMPFunction, 23
- SNMPLocation, 23
- SNMPStation, 23
- SyncModeDHCPD_SysNet, 19
- Time zone, 26
- Timeservers, 17

## C

CFS (Compressed File System), 40
Command
- Backup, 54
- Check-cf, 43
- Clear, 43
- Closeport, 44
- Connect, 44
- Disable, 47
- Disconnect, 48
- Enable, 49
- Help, 42
- ntpdiag, 50
- Openport, 51
- Port status, 52
- Reboot, 52
- Restart, 52
- Restore, 53
- Save, 54
- save-nvram, 55
- set-svc-route, 56
- Show, 56
- Start, 60
- Stop, 60
- usb, 61
- vncpwd, 62

CompactFlash Card
- Memory partitioning, 6
- Partitions, 5

## D

data
- Restoring, 82

DHCP synchronization, 50

## E

Emergency Boot System, 63
Ethernet interfaces, 13

## I

Interfaces
  Ethernet, 40
  NCU, 13
IP address
  Display, 14

## L

Load operating system, 10

## N

NCK commissioning switch, 8
NCU booting, 9
Network drive, 67
Network interface, 13
NFS (Network File System), 41

## P

PLC mode switch, 8
Power on reset, 9
Privilege, 38

## R

Remote File System, 41
Replacement part, 69
Reset button, 9

## S

Section
  [DCP], 24
  [ExternalInterface], 16
  [IBNInterface], 21
  [InternalInterface], 18
  [LinuxBase], 25
  [LLDP], 24
  [SNMP], 22
Service command
  Application, 38
  Rights, 38
  Syntax, 39
Service menu, 66
Service system
  Backing up data, 80
  Creating for the NCU, 64
  Restoring data, 81
Service Tool WinSCP, 96
SMB (Server Message Block), 41
Subnet mask, 14
Subsystem, 40
  Designation, 40
Synchronization
  DHCP, 50
System
  Diagnostics, 10
  Log file, 5, 29
System error, 12

## T

Time zone, 26

## U

Update
  BIOS, 84
  Firmware, 84
User group, 38
Users (default setting), 6

## V

VNC (Virtual Network Computing), 13

## W

WinSCP, 96

SIEMENS

# SINUMERIK

# SINUMERIK 840D sl PCU-Basesoftware (IM8)

Commissioning Manual


Commissioning the system 1

Configuring the system 2

Install software and updates 3

Backing up and restoring data 4

Service and diagnostics 5

List of Abbreviations A


Valid for:

SINUMERIK PCU 50.5
with PCU base software V5XP1.4

SINUMERIK PCU 50.3
with PCU base software V8.6 SP3


02/2014
6FC5397-1DP40-6BA2

## 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

Document order number: 6FC5397-1DP40-6BA2
℗ 08/2018 Subject to change

Copyright © Siemens AG 2012 - 2014.
All rights reserved

# Table of contents


**1 Commissioning the system....5**

1.1 Delivery condition of the system....5
1.1.1 Hard disk partitions....6
1.1.2 System features....7

1.2 User administration....9
1.2.1 Which users are set up?....9
1.2.2 User settings....10
1.2.3 This is how you create a new user....17
1.2.4 This is how you administer users....18

1.3 Boot up behavior of the PCU....19
1.3.1 Dialogs when booting....19
1.3.2 Boot up of the PCU: no HMI program installed....21
1.3.3 Boot up of the PCU: HMI program already installed....23
1.3.4 Setting the screen resolution....25

1.4 BIOS settings for PCU 50.3....28

1.5 BIOS settings for PCU 50.5....33

**2 Configuring the system....39**

2.1 System settings....39
2.1.1 Configuring the communication via MPI....39
2.1.2 How to change the name of the PCU....39
2.1.3 This is how you set the IP address of the PCU....40
2.1.4 How to add the PCU to a domain....42
2.1.5 How to connect an external monitor....43
2.1.6 This is how you switch off the USB interfaces....44

2.2 Configuring a customized operator interface....45
2.2.1 How to select the language for the Windows system....45
2.2.2 Displaying the boot screen....47
2.2.3 Changing the background of the service desktop....48

2.3 Customized settings during boot up....49
2.3.1 Configuring key filters for an HMI program....49
2.3.2 Saving service desktop settings....52
2.3.3 Starting programs during boot up....53
2.3.4 Starting OEM programs....54
2.3.5 Starting applications in service mode....54

2.4 Configuring the SITOP UPS module for use with the PCU Base Software....56
2.4.1 SITOP modules for the PCU 50.5....56
2.4.2 Configuring SITOP software for PCU Base Software....57
2.4.3 Parameterizing a delay time for quitting the HMI software....60
2.4.4 Hardware configuration of the SITOP UPS module....62

2.5 Configuration of the Service Center....65
2.5.1 Overview....65

2.5.2 Configuration of the network adapter....66
2.5.3 Configuration of the host....67
**3 Install software and updates....69**
3.1 Installing SINUMERIK products....69
3.2 Installation via service desktop....70
3.3 How to install and authorize SIMATIC STEP 7....73
3.4 How to install additional languages under Windows XP (DVD)....74
**4 Backing up and restoring data....77**
4.1 Overview....77
4.2 Starting the Service Center....78
4.3 Functions of the Service Center....79
4.4 Network settings in the Service Center....81
4.5 Create a disk image of the SSD....84
4.6 Restore a disk image of the SSD....86
4.7 Create a disk image of a partition....88
4.8 Restore a disk image of a partition....90
4.9 Network settings in the Service Center....93
4.10 Starting Symantec Ghost directly....95
4.11 Bootable USB flash drive....96
4.12 Operating a service PC/PC in the network....97
4.12.1 Overview....97
4.12.2 Connection options in the system network....97
4.12.3 Configuring routing in the network....99
4.12.4 Configure the network settings of the PG/PC....101
4.12.5 Release directory of the PC/PC in the network....103
4.12.6 Release drives C: and D: of the PCU 50.5 in the network for administrator accounts....106
4.12.7 Creating a shortcut to the network drive....107
4.13 Starting up the replacement SSD....109
**5 Service and diagnostics....111**
5.1 PCU Hardware Diagnostics....111
5.2 Evaluation of the 7-segment display....112
5.3 Setting of the service switch (PCU 50.5 only)....114
5.4 Enabling/disabling error log during boot up....115
5.5 How to search for stations within the system network....116
**A List of Abbreviations....121**
A.1 Abbreviations....121
**Index....123**

# Commissioning the system

# 1

## 1.1 Delivery condition of the system

### Overview

The PCU has onboard interfaces for communicating via Ethernet, MPI and PROFIBUS DP. The integrated free slots remain free for other tasks. The PCU is equipped with the Windows XP ProEmbSys operating system and for data backup tasks with the Symantec Ghost software.

Interfaces:

- Four USB ports (USB 2.0) offer points where a keyboard, mouse and other peripheral devices can be connected.
- For CF cards, there is a covered slot.
- Two internal PCI slots are available for specific expansions.

For commissioning:

- Two 7-segment displays and two LEDs are integrated for diagnostic purposes. They indicate the current operating status and display the BIOS error codes during startup.
- If the PCU is to be operated without an operator panel front, a monitor and an additional keyboard will also be required:
  - For diagnostics during startup of the PCU.
  - When installing a replacement hard disk (alternatively, the hard disk can also be installed externally).

**References:** Operator Components and Networking Manual

### Supplied software on the PCU

The software installed on delivery of the PCU includes the components below, among others:

| **MS Windows XP Professional SP3** | |
|---|---|
| Internet Explorer | V 6.0 |
| MPI driver (PCU 50.3 and PCU 50.5-C) | V 6.04.01.00 |
| Symantec Ghost (default setting) | V 8.2 (incl. Ghost Explorer) |
| TCU Support<br>(It is already installed and available on the hard disk under D:\Updates, if it must be re-installed.) | V 8.6 |

Documentation for all Ghost tools is supplied on the PCU's hard disk under E:\TOOLS

**Note**

For the system component versions contained in the PCU base software, see the C:\BaseVers.txt file.

**Note**

**Acknowledgements**

- This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)
- This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)

## 1.1.1 Hard disk partitions

### Memory segmentation

The hard disk of the **PCU 50.3** has 40 GB of storage capacity and is segmented into a primary partition C and an expanded partition with the three logical drives D, E and F which operate with NTFS file access.

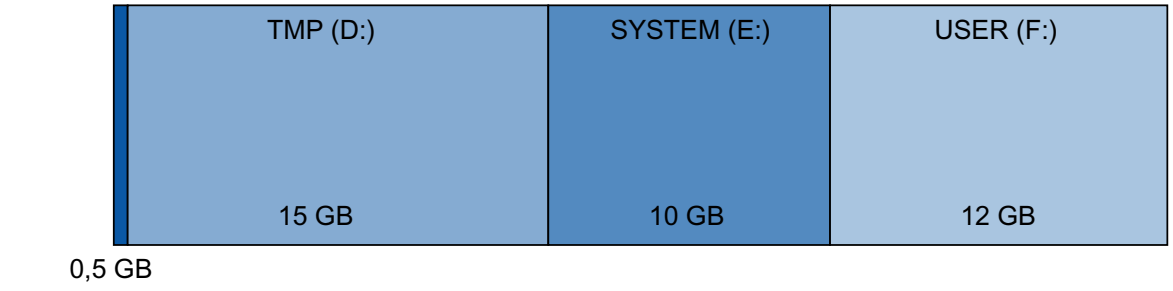



Figure 1-1 Division of the hard disk

The **PCU 50.5** is equipped with a solid state drive (SSD) with NTFS file access, which is also segmented to provide a primary partition C and an expanded partition with the three logical drives D, E and F. The SSD is available with 32 GB and 40 GB storage capacity.

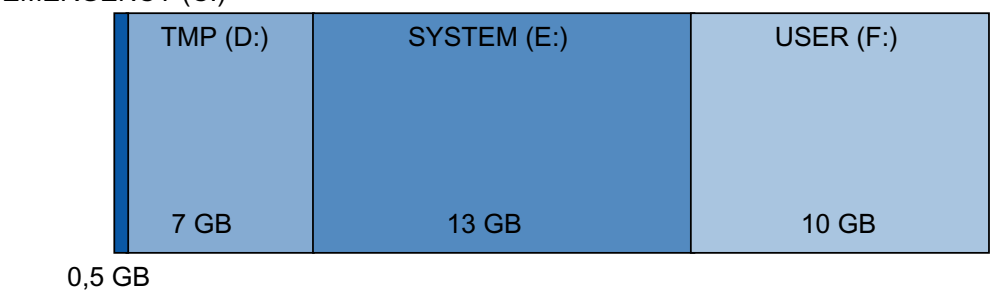



Figure 1-2 Segmentation of the SSD with 32 GB

Partitioning of the SSD with 40 GB is identical to that for partitioning of the hard disk.

Depending on the order, the operating software can already be installed when the unit is shipped. It is installed subsequently by the customer the first time the system starts up. For reasons of data security, the operating software and the Windows XP system software are distributed over the different hard disk partitions.

### Content of the partitions

The individual partitions are intended for the following data or already contain this data:

| | |
|---|---|
| EMERGENCY (C:) | Reserved for service tasks under WinPE 2005. |
| TMP (D:) | Used for storing Ghost images (e.g. of the delivery condition) and local backup images.<br>Contains the installation directory where the software to be installed is first copied to from a remote PG/PC prior to the actual installation procedure. |
| SYSTEM (E:) | Reserved for the Windows XP software.<br>The Windows XP software is available on the recovery media CD via network, for example to install drivers or updates as and when needed later on. |
| USER (F:) | For installing user programs.<br>Applications such as HMI system software (incl. data storage and temporary data), STEP 7, OEM applications for HMI or customer-specific applications should only be installed here. |

**Note**

**Installing software**

All of the applications must be exclusively installed on USER (F:) even if these applications have a different drive set as the default drive in their installation path.

The partition names EMERGENCY, TMP, SYSTEM, USER must not be changed; otherwise the "ServiceCenter" will no longer function.

### See also

Installing SINUMERIK products (Page 69)

## 1.1.2 System features

### Configuration of the operating system

For safety reasons, Windows XP has been preset as follows:

- The Autorun function is deactivated.
- Automatic Windows Update is deactivated.

- Monitoring and alerts for antivirus software and automatic update are deactivated.
- Links used to call up Internet Explorer from the service desktop and the start menu are removed.
- Remote Procedure Call (RPC) is possible for calls that are not connected.
- The firewall settings are activated on the network card Eth 1 and deactivated on Eth 2.

### Changes to Windows services

Other default settings:

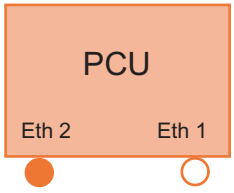
| Windows services: | Start-up type: | |
|---|---|---|
| Computer Browser | Manual | (Not started) |
| Error Reporting Service | Disabled | |
| Portable Media Serial Number | Manual | (Not started) |
| SSDP Discovery Service | Disabled | |
| Universal Plug and Play Host | Disabled | |
| Web Client | Manual | (Not started) |
| Wireless Zero Configuration | Manual | (Not started) |

### Name of the PCU

Upon delivery of the system, a unique computer name is generated which can be read out under: "Start" → "Settings" → "Control Panel" → "System", "Computer Name" tab.

### Pre-configuration of the PCU

The PCU has two Ethernet interfaces with default settings suitable to connect to SINUMERIK solution line:




**Eth 1** is preset as a default DHCP client for connection to a company network.

**Eth 2** is preset as a SINUMERIK DHCP server for connection to a system network. Eth 2 is preset to the fixed IP address 192.168.214.241.

### See also

This is how you set the IP address of the PCU (Page 40)

**References:** Operator Components and Networking Manual

# 1.2 User administration

## 1.2.1 Which users are set up?

### Preset users

Each user is from one user type and belongs to one user group. The user types are implemented under Windows in user groups with various user rights.

Upon delivery of the system, the following users are defined at the factory:

- **The "operator"**
  In terms of type, the "operator" is classed as an HMI user and belongs to the operators' user group (operator.group). These operators have limited user rights under Windows.
- **The "user"**
  In terms of type, the "user" is classed as an HMI user and belongs to the operators' user group (user.group). These operators have extended user rights under Windows.
- **The "auduser".**
  The "auduser" is an HMI service user type of user and belongs to the system administrators' user group. The system administrators have the user rights of a local administrator under Windows.

| User name | Password | User type | Windows user group | User Rights | User group |
|---|---|---|---|---|---|
| operator | operator | HMI (operator.group) | operator.group | Restricted | Operator |
| user | CUSTOMER | HMI (user.group) | user.group | "Power User" | Operator |
| auduser | SUNRISE | HMI+Service | Administrators | local administrator | System administrators |
| siemens | ***** | --- | --- | --- | System administrators |

The individual user types differ in the following areas:

| User type | Area of activity |
|---|---|
| HMI (operator.group) | • Boot up of the PCU<br>• Operating the HMI program<br>• Windows Desktop |
| HMI (user.group) | • Boot up of the PCU<br>• Operating the HMI program<br>• Windows desktop |
| HMI+Service | • Boot up of the PCU<br>• Operating the HMI program<br>• Service Desktop |

### Service Desktop

The service desktop provides the HMI+Service user with a Windows desktop, which is expanded to include tools and functions for service tasks, such as manage users, install software, save/restore data, check system integrity, etc.

### Windows desktop

The HMI user is able to use a Windows desktop which can be expanded to suit his or her individual needs.

### Starting ServiceCenter Users




Using this link on the service desktop, you can start ServiceCenter Users **as an HMI+Service user**. In ServiceCenter Users, the boot behavior of the PCU as well as settings for the HMI program and desktop are set globally for all users and individually for specific users.

The users are managed in the "ServiceCenter Users", so that the commissioner/service technician no longer has to make corresponding settings directly in the registry.

## 1.2.2 User settings

### "Global Settings"

In the ServiceCenter Users, as **HMI service user**, you set parameters under "Global Settings" for the system behavior:

- When the PCU is booting up
- When the HMI program is starting
- With reference to the desktop

**Note**

The following diagrams show, as an example, an application with the SINUMERIK Operate software installed.

### Using ServiceCenter Users

Buttons:

- Use "Exit" to quit ServiceCenter Users.
- If you press the "Reboot" button, the system immediately reboots (without prior prompting).

## "Startup (1)" tab

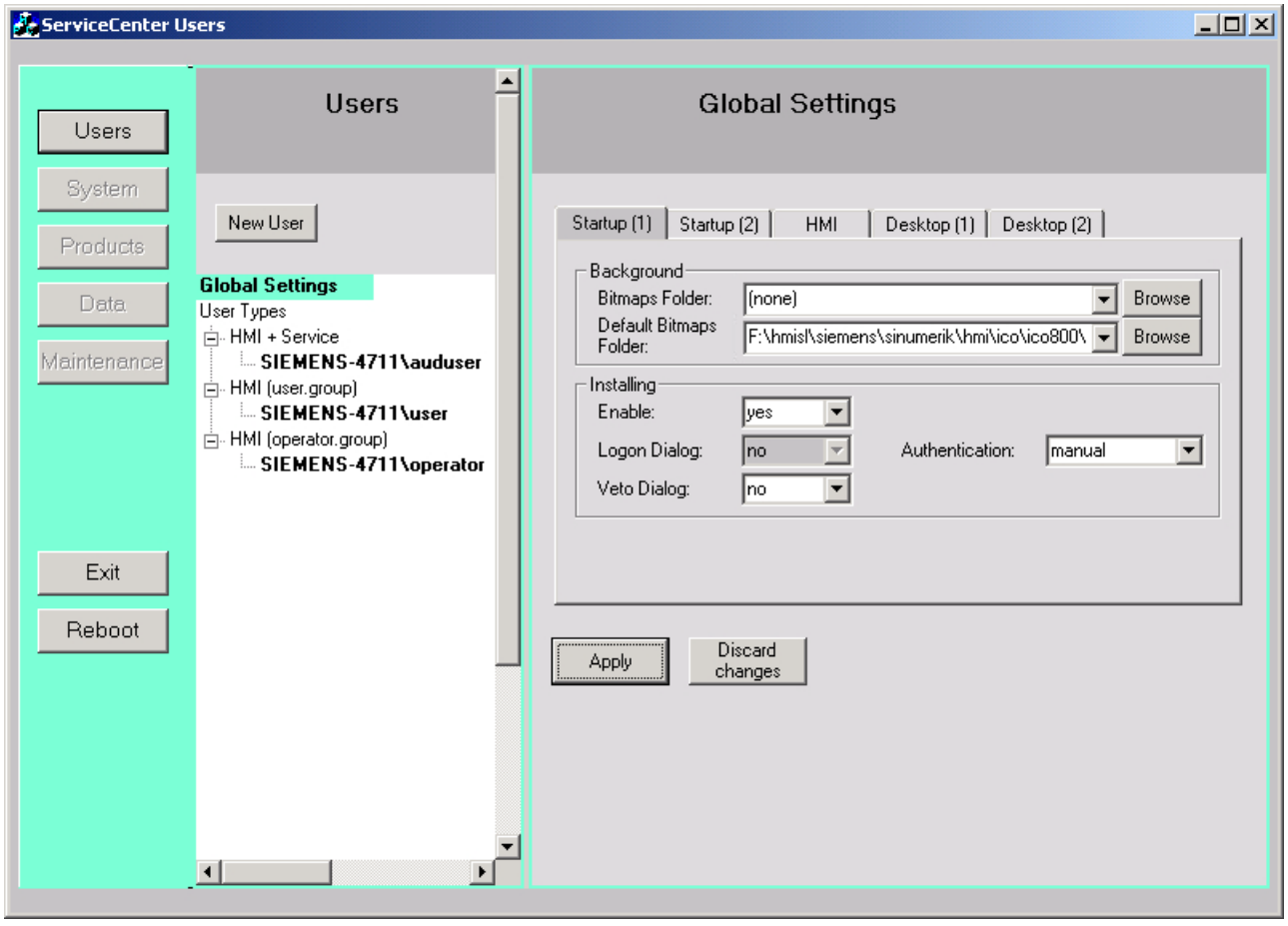



Figure 1-3 Global Settings: Startup (1) tab

Table 1-1 The default is marked in "**bold**".

| **Startup (1) tab** | **Option** | **Effect** |
|---|---|---|
| **Background** | | |
| Bitmaps folder: | F:\hmisl\siemens\sinumerik\hmi\ico\ ... | Directory with boot screens |
| Default bitmap folder: | **F:\hmisl\siemens\sinumerik\hmi\ico\ico800** | Directory containing default boot screens |
| **Installing** | | |
| Enable: | **"yes"** | Authorization is in place to install user software during booting. |
| | "no" | No installation authorization |
| Logon dialog: | "yes" | Display "Installing Logon" dialog. |
| | **"no"** | Do not display "Installing Logon" dialog. |
| Veto Dialog: | **"no"** | For pending installation during the boot up: No prompting, the installation begins immediately. |

| Startup (1) tab | Option | Effect |
|---|---|---|
| | "yes" | For pending installation during the boot up: Prompt asking whether to install. |
| Authentication: | **"manual"** | Authentication of an HMI+Service user in the "Installing Logon" dialog is undertaken manually. |
| | "EKS" | Authentication of an HMI+Service user in the "Installing Logon" dialog involves EKS. The "Installing Logon" dialog appears if authentication by EKS has failed in the background. |
| | "manual+EKS" | Authentication of an HMI+Service user in the "Installing Logon" dialog either involves EKS or is performed manually. |

## "Startup (2)" tab

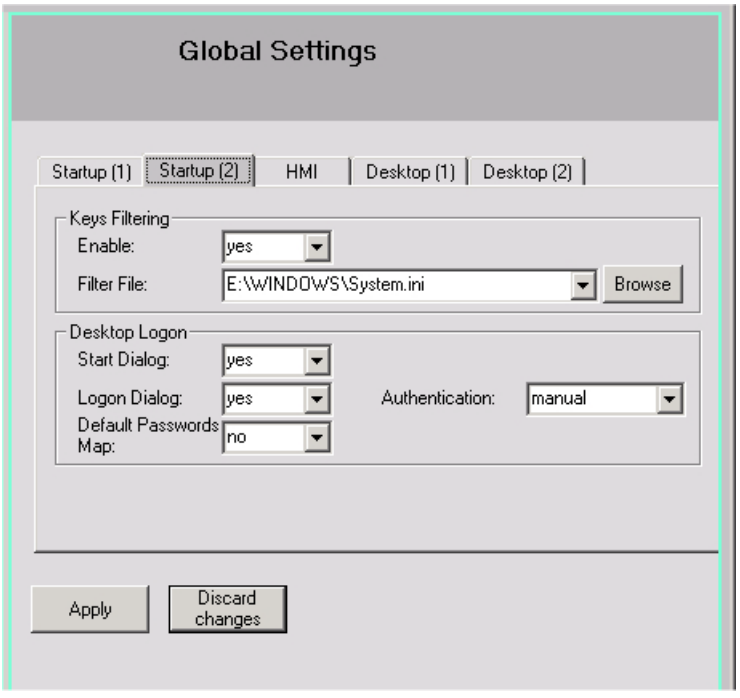



Figure 1-4 "Startup (2)" tab

Table 1-2 The default is marked in "**bold**".

| Startup (2) tab | Option | Effect |
|---|---|---|
| **Keys, filtering** | | |
| Enable | **"yes"** | Key filtering is activated. |
| | "no" | Key filtering not activated. |
| Filter file | **E:\WINDOWS\System.ini** | Select the file with filter settings |
| **Desktop Logon** | | |
| Start dialog: | **"yes"** | Display "Desktop Access" dialog |

| Startup (2) tab | Option | Effect |
|---|---|---|
| | "no" | Do not display "Desktop Access" dialog |
| Logon Dialog: | **"yes"** | Display "Desktop Logon" dialog. |
| | "no" | Do not display "Desktop Logon" dialog. |
| Default password map: | **"no"** | The password must be entered exactly as specified. |
| | "yes" | The password is not case-sensitive. |
| Authentication: | **"manual"** | Authentication with user name and password (manual) |
| | "EKS" | Authentication with EKS |
| | "manual+EKS" | Either authentication with user name and password or with EKS |

## "HMI" tab

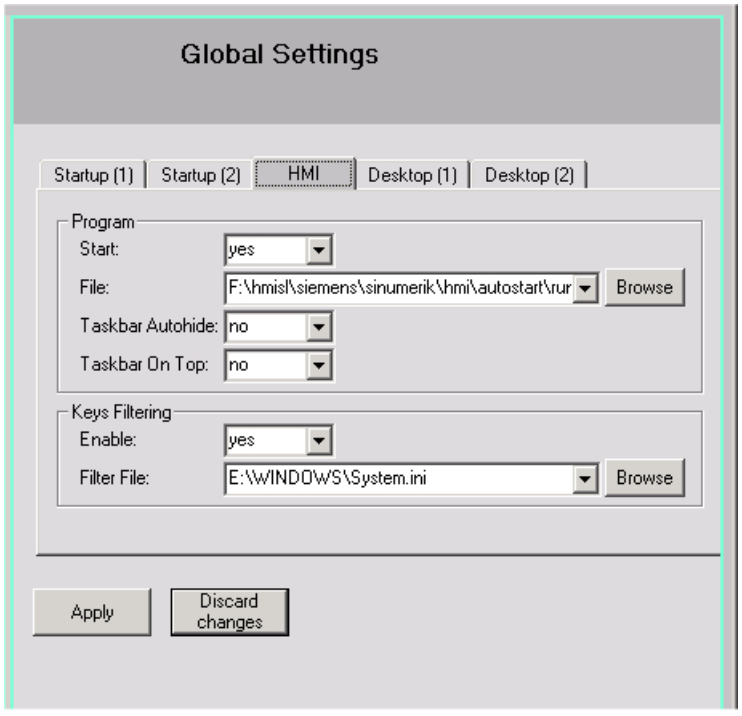



Figure 1-5 "HMI" tab

Table 1-3 The default is marked in "**bold**".

| "HMI" tab | Option | Effect |
|---|---|---|
| **Keys Filtering** | | |
| Enable | **"yes"** | Key filtering is activated. |
| | "no" | Key filtering not activated. |
| Filter file | **E:\WINDOWS\System.ini** | Select the file with filter settings |
| **Program** | | |
| Start: | **"yes"** | The HMI program is started. |

| **"HMI" tab** | **Option** | **Effect** |
|---|---|---|
| | "no" | The HMI program is not started. |
| File: | **F:\hmisl\siemens\sinumerik\hmi\autostart\run_hmi.exe** | Select the HMI program |
| Task bar autohide: | **"no"** | HMI program: Hide start task bar |
| | "yes" | HMI program: Show start task bar |
| Task bar on top: | **"no"** | HMI program: Start task bar in the background |
| | "yes" | HMI program: Start task bar always visible |

## "Desktop (1)" tab

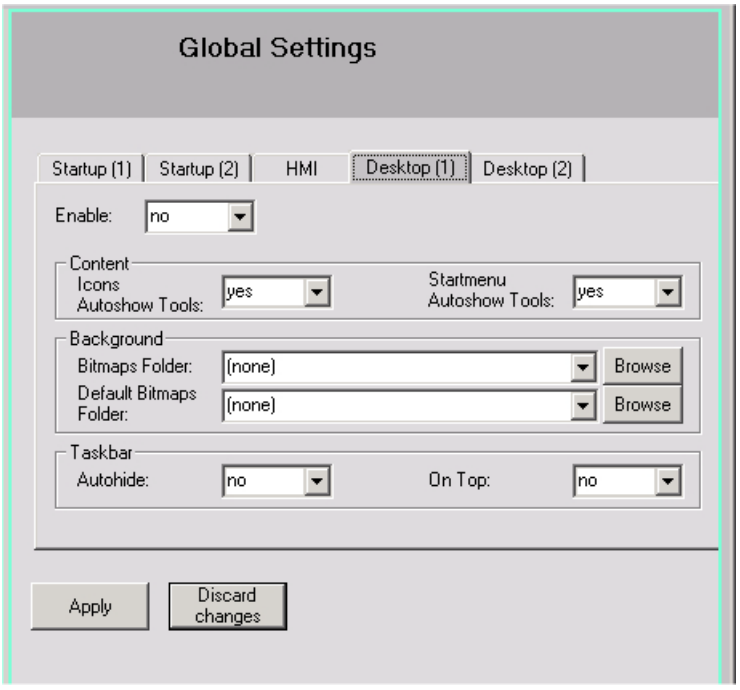



Figure 1-6 "Desktop (1)" tab

Table 1-4 The default is marked in "**bold**".

| **"Desktop (1)" tab** | **Option** | **Effect** |
|---|---|---|
| Enable: | "no" | Desktop not displayed<br>**(Default: for HMI user)** |
| | "yes" | Displaying the desktop<br>**(Default: only for HMI+Service user)** |
| **Content** | | |
| Icons Autoshow Tools: | "no" | Do not display links on the desktop. |
| | **"yes"** | Display links on the desktop. |
| Startmenu Autosh-ow Tools: | "no" | Do not display programs in the "Start menu" |
| | **"yes"** | Display programs in the "Start menu". |

| "Desktop (1)" tab | Option | Effect |
|---|---|---|
| **Background** | | |
| Bitmaps folder: | **(none)** | Directory with background screens for the desktop |
| Default bitmap folder: | **(none)** | Directory with default background screens for the desktop |
| **Task bar** | | |
| Autohide: | **"no"** | Desktop: Hide start task bar |
| | "yes" | Desktop: Show start task bar |
| On Top: | **"no"** | Desktop: Start task bar in the background |
| | "yes" | Desktop: Start task bar always visible |

## "Desktop (2)" tab

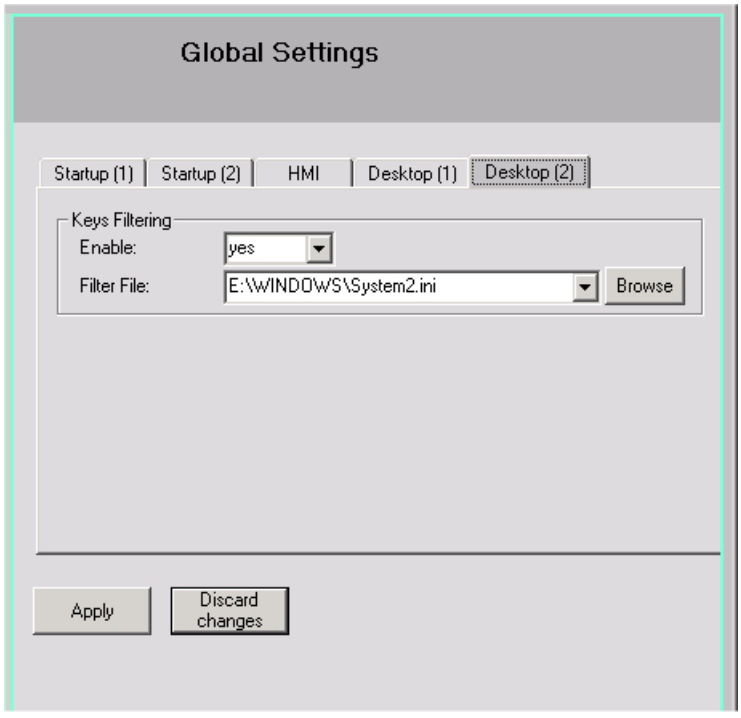



Figure 1-7 SCU "Desktop (2)" tab

Table 1-5 The default is marked in "**bold**".

| "Desktop (2)" tab | Option | Effect |
|---|---|---|
| **Keys, filtering** | | |
| Enable | **"yes"** | Key filtering is activated. |
| | "no" | Key filtering not activated. |
| Filter file | **E:\WINDOWS\System2.ini** | Select the file with filter settings |

## Principle of passing on

The settings made under "Global Settings" are passed on to individual users. The settings which are passed on can still be adapted to suit each user.

## Example: Specific user settings

The settings under "Global Settings" are inherited by all users:

1. Left-click the user in the "Users" list.
2. After entering the password, you can adapt the settings as required.
   The settings that cannot be changed are displayed with a gray background.

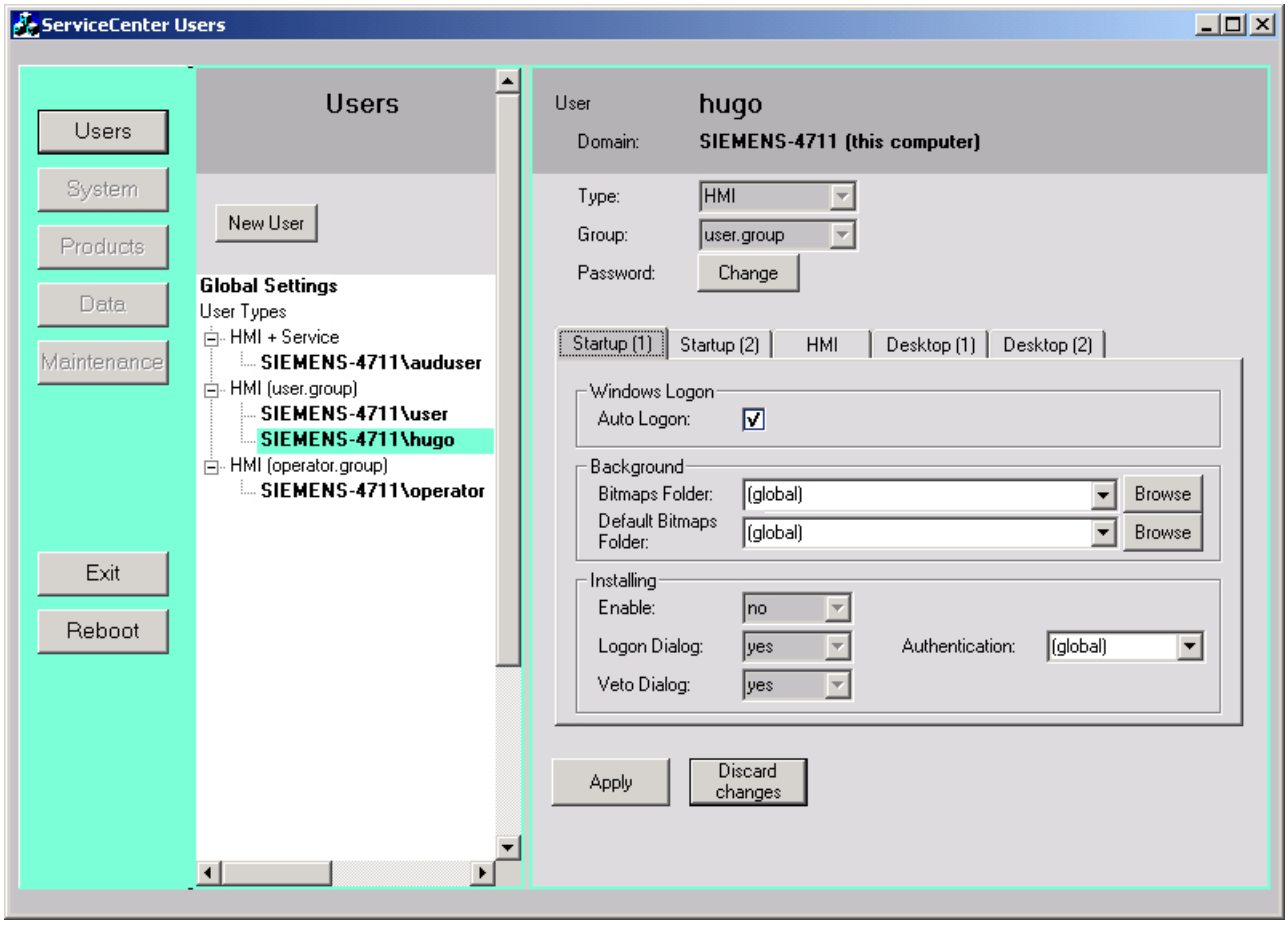



Figure 1-8 Specific settings for the "HMI (user.group)" user

## 1.2.3 This is how you create a new user

### Creating new users

Procedure:

1. Click on "New User" button.
2. Enter a user name.
3. Assign a user type, e.g. "HMI", and a user group to the user.
   The Windows user group is only active locally on the PCU.

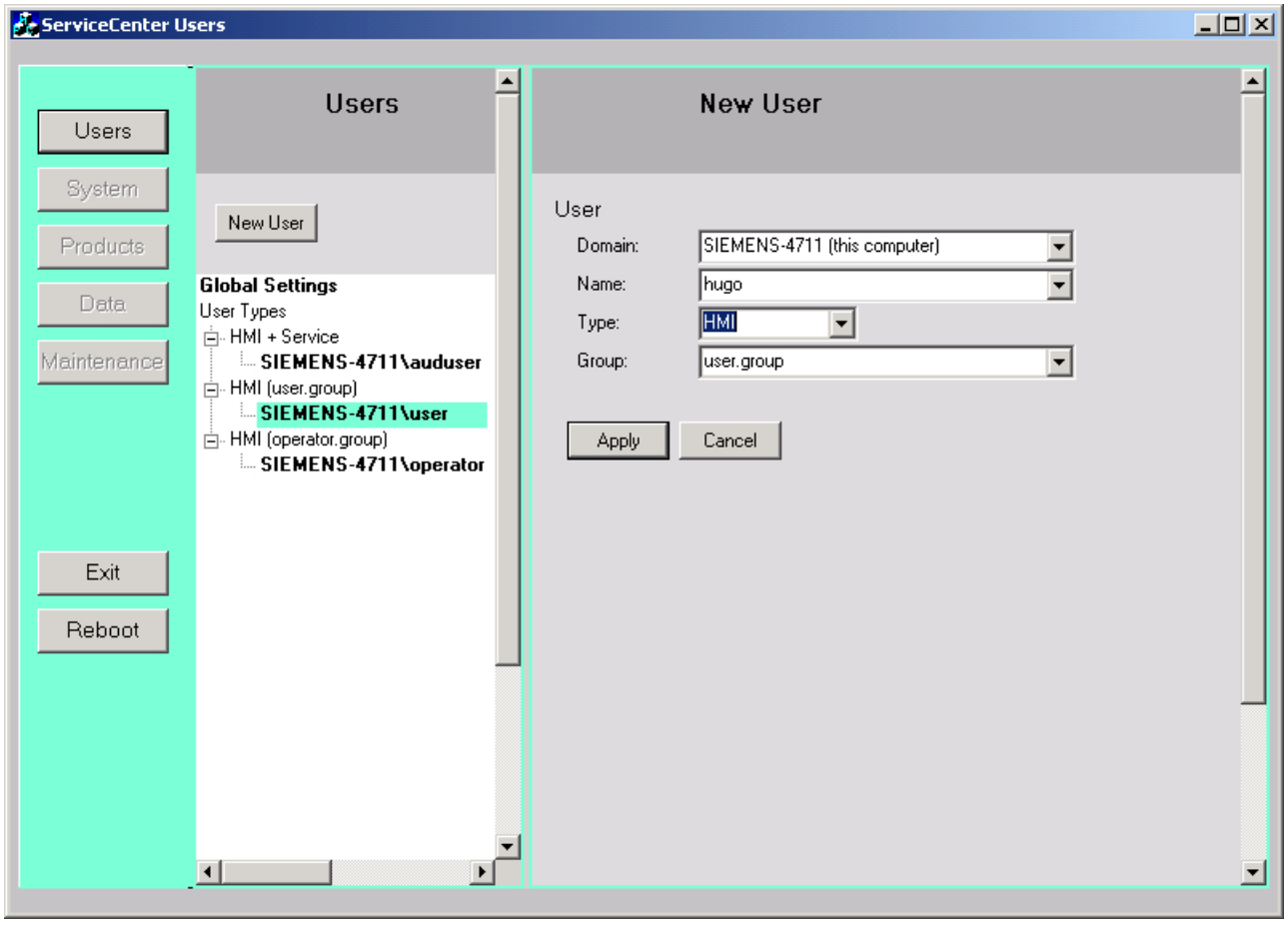



Figure 1-9 ServiceCenter Users: New user

4. Confirm with "Apply".
5. You will then be asked to specify a password.
6. After confirming with "OK", the new user is created and displayed in the list under "Users".

### Adding new users from a domain

If the PCU is a member of a domain, users already existing in this domain can be added as HMI users or HMI+Service users:

1. Click on the "New User" button.
2. To do this, select the corresponding entry in the "Domain" list and a user from the "Name" list.
3. Assign a user type and a user group to the user.
   The Windows user group is only active locally on the PCU.

### See also

How to add the PCU to a domain (Page 42)

## 1.2.4 This is how you administer users

### Overview

As an HMI+Service user, you can execute the following tasks in the Users ServiceCenter:

- Change user names.
- Delete users.

### Change the user name

1. Left or right-click the user who you wish to rename in the list.
2. To change the user name, select "Rename" from the pop-up menu.
3. Enter a new name and confirm with OK.

### Deleting a user

To delete a user again, follow these steps:

1. Left or right-click the user who you wish to delete in the list.
2. To delete the user, select "Delete" from the short-cut menu using righthand mouse key.

**Result:** The user is deleted after confirming the confirmation prompt.

# 1.3 Boot up behavior of the PCU

## 1.3.1 Dialogs when booting

### Selection during startup

The following dialogs are shown during startup:

- **"Installing Logon" dialog**
  This dialog is displayed if an "Install" directory is found on a bootable storage medium (e.g. USB-FlashDrive) during startup.

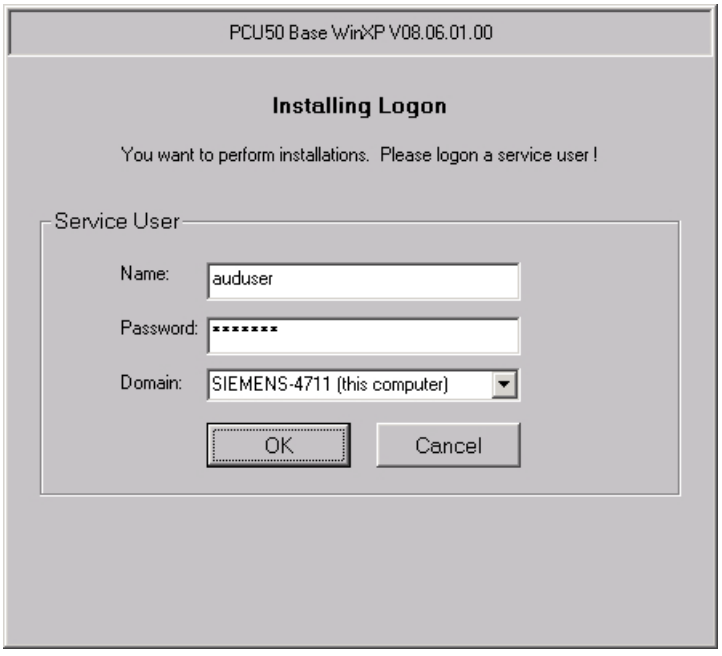



Figure 1-10 Installing Logon

- **"Desktop Access" dialog**
  - This dialog is displayed if **either** no HMI program is installed, the installation has been skipped **or** the start of the HMI program has been deactivated during startup. Displaying this dialog can be suppressed through configuration in the global or user settings. "Desktop" is then selected as the default.

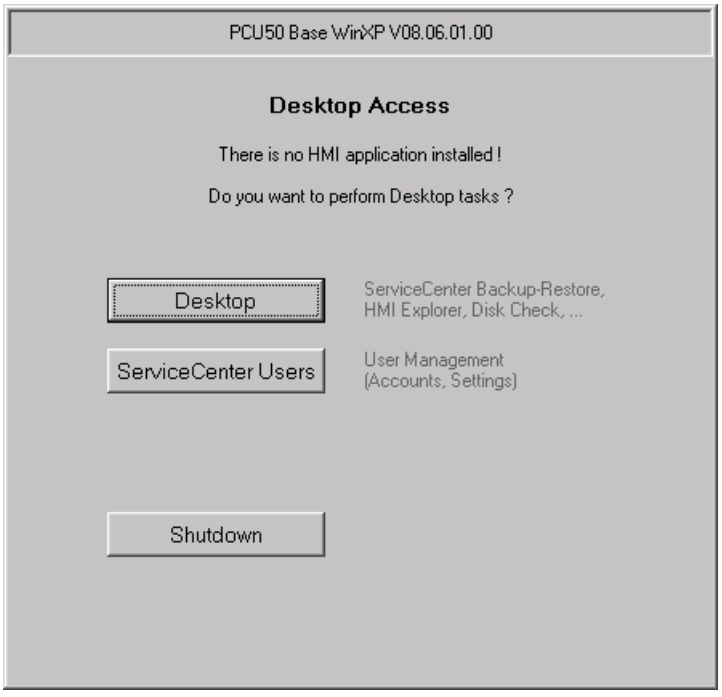



Figure 1-11 Desktop Access (example without HMI program)

**Note**

If an HMI program is installed, the "Start HMI" button is also available in the "Desktop Access" dialog.

- **"Desktop Logon" dialog**
  The "Desktop Logon" dialog enables the user to log on to the Desktop. This dialog also appears
  - If no valid user data has been provided by the authentication system (EKS).

– If authentication on the "Startup(1)" tab is set to "manual".

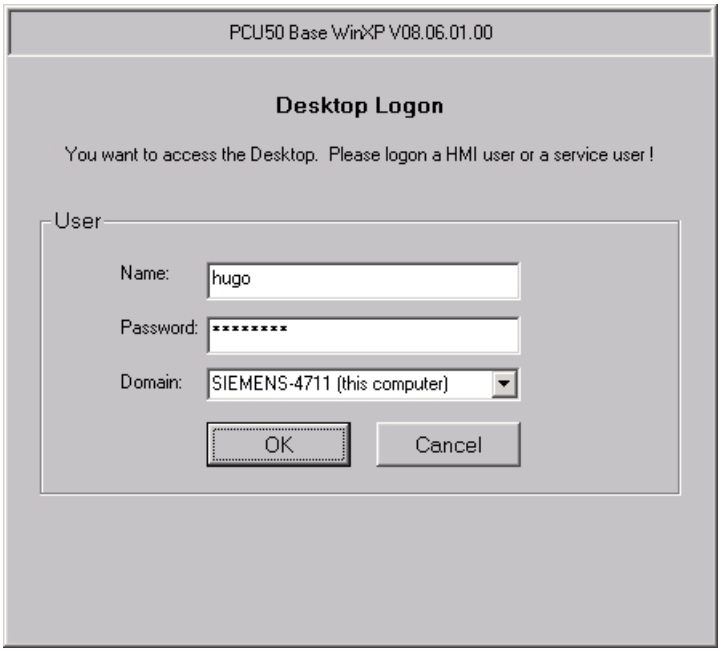



Figure 1-12 Desktop Logon (with authentication: "manual")

**Note**

**Setting the access level via EKS**

If an EKS unit is available, the following behavior applies:

- If an EKS unit is active, i.e. the key is inserted and can be evaluated, then the key information for the EKS unit alone determines the active access level. The access level in the HMI program cannot be changed when EKS is active.
- If the key is withdrawn from the active EKS unit, i.e. the EKS unit becomes inactive, the system adopts the current access level as determined by the key switch.
- If changing the operating right amongst the operating stations, the EKS unit of the new active operating station is authoritative. If no EKS unit is assigned there, the effect is the same as for an inactive EKS unit.
- The key information is evaluated by the HMI. Once an access level has been determined from the key information, HMI adopts this access level and also sets it in the NCK: The last access level set always applies to the system.

## 1.3.2 Boot up of the PCU: no HMI program installed

### Requirement

No HMI program is installed during the PCU boot up.

## Ramp-up phase

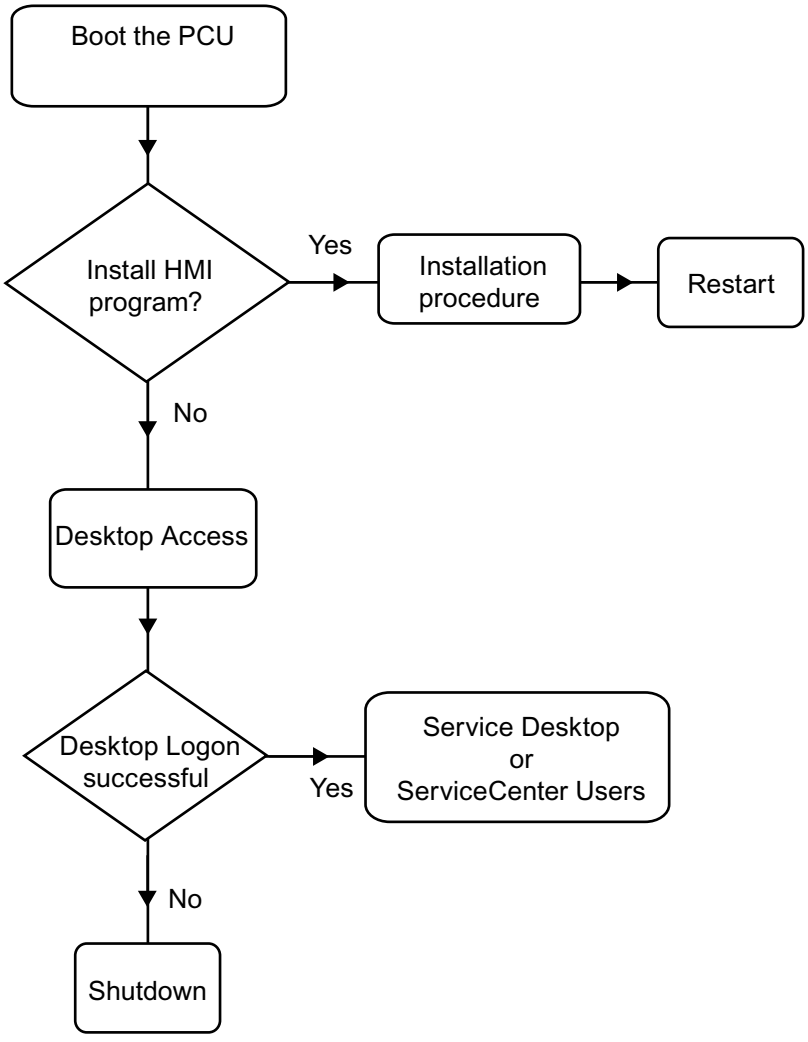



Figure 1-13 PCU bootup diagram (without HMI program)

Explanations:

- If installation programs are present in directory D:\Install, you will be prompted during the first boot up as to whether the installation procedure should be started. After installation is completed, you need to restart the system.
  The installation can also be skipped and carried out later. To do this, use the HMI Explorer (Installation via service desktop (Page 70)).
- "Desktop", "ServiceCenter Users" or "Shutdown" can be selected in the dialog "Desktop Access".
  - When selecting "Desktop", the "Desktop Logon" dialog is opened.
  - When selecting "ServiceCenter Users", the "Installing Logon" dialog is opened.

  **Note**

  When booting the PCU for the **first** time, the user can only log on as 'auduser'.

### 1.3.3 Boot up of the PCU: HMI program already installed

#### Requirement

There is already an HMI program installed.

## Ramp-up phase

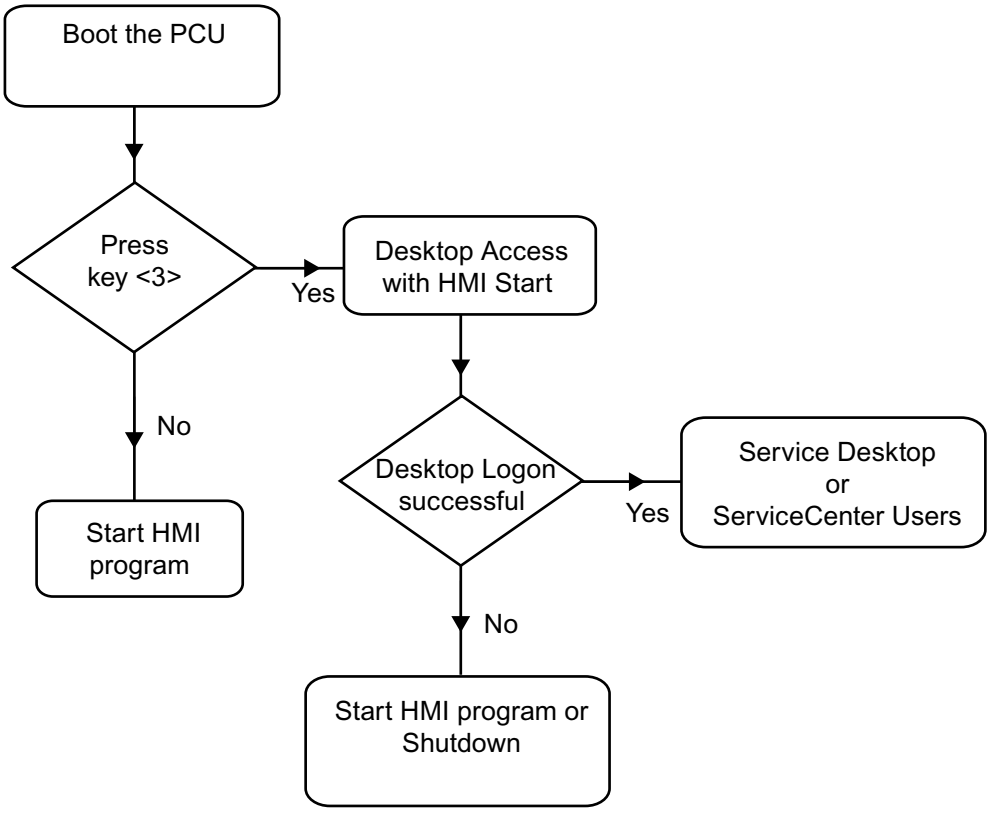



Figure 1-14 PCU bootup diagram (HMI program installed)

Explanations:

- If the HMI program is already installed, the PCU boots up and the HMI program is started (default).
- During the boot up, there is a time interval for pressing key <3> when the version information appears on the lower right of the background screen. Then the "Installing Logon" dialog opens.
- To carry out service tasks, you will have to log on as a service user.
  The following input options are available:
  - Manual logon to a domain using user name and password.
  - Logon using EKS: A key and valid user data for authentication must be provided for this. If valid user data is not available via the EKS, the "Desktop Logon" dialog is displayed along with user name and password.
  - Both options can be selected.

## 1.3.4 Setting the screen resolution

### Overview

The system behavior during boot up for the screen resolution is set in the file tcu.ini. You will find the delivery status for tcu.ini in E:\siemens\system\etc.

Modified tcu.ini files are saved in F:\addon_base\..., F:\oem_base\..., F:\user_base\...

**Reference:** Operator Components Manual, "Networking" chapter

### Set the resolution when booting up the PCU

The following options are available in the # RESOLUTION section in the tcu.ini:

0 = SYSTEM

1 = AUTO_OP_1 (default)

2 = AUTO_OP_2

3 = AUTO_MON_1

4 = AUTO_MON_2

5 = 640X480

6 = 800X600

7 = 1024X768

8 = 1280X1024

The meanings of the settings are as follows:

| Settings | Meaning |
|---|---|
| SYSTEM | The resolution is not specially set; i.e., the resolution last used in the system is active, e.g., the resolution which had been set manually in the Control Panel. |
| AUTO_OP_1 | Default:<br>During boot up, the resolution is automatically set ("PCU panel" has priority) in accordance with the following scenarios: |
| Example 1: | There is a PCU panel (irrespective of whether there is a PCU monitor and TCU panels)<br>[in active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is set to the max. resolution of the PCU panel (max. 1280x1024). |

| Settings | Meaning |
|---|---|
| Example 2: | There is no PCU panel, however there is a PCU monitor (irrespective of whether there are any TCU panels):<br>[in active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is not specially set; i.e., the resolution last used in the system is active, e.g., the resolution which had been set manually in the Control Panel.<br>(Different to AUTO_OP_2 !) |
| AUTO_OP_2 | Like AUTO_OP_1, except:<br>Example 2: There is no PCU panel, however there is a PCU monitor (irrespective of whether there are any TCU panels):<br>[in active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is set to the max. resolution of the PCU monitor, reduced to the next lowest SINUMERIK resolution. The SINUMERIK resolutions are 640x480, 800x600, 1024x768 and 1280x1024.<br>**Example:** In the case of a PCU monitor with a max. resolution of 1440x900, the SINUMERIK resolution setting is 1280x1024. |
| AUTO_MON_1 | During boot up, the resolution is automatically set ("PCU monitor" has priority) in accordance with the following scenarios: |
| Example 1: | There is a PCU monitor (irrespective of whether there is a PCU panel and TCU panels)<br>[in active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is set to the max. resolution of the PCU monitor, reduced to the next lowest SINUMERIK resolution. The SINUMERIK resolutions are 640x480, 800x600, 1024x768 and 1280x1024.<br>**Example:** In the case of a PCU monitor with a max. resolution of 1440x900, the SINUMERIK resolution setting is 1280x1024.<br>If there is a PCU panel, the display there is panned if the max. resolution of the PCU panel is lower than the max. resolution of the PCU monitor. |
| Example 2: | There is no PCU monitor, however there is a PCU panel (irrespective of whether there are any TCU panels):<br>[in active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is not specially set; i.e., the resolution last used in the system is active, e.g., the resolution which had been set manually in the Control Panel.<br>(Different to AUTO_MON_2 !) |

| Settings | Meaning |
|---|---|
| Example 3: | There is no PCU monitor and no PCU panel (= headless operation):<br>A) in active TCU mode:<br>a) dynamic resolution switching is enabled (resolution adaptation entry in TCU.ini) and at least one TCU is already logged on:<br>The resolution is set to the resolution of the TCU which is currently active.<br>b) dynamic resolution switching is enabled (resolution adaptation entry in TCU.ini) and no TCU has logged on yet or dynamic resolution switching is disabled:<br>The resolution is set to the max. resolution of the current PCU panel, i.e., which logged on during the previous session.<br>Default: Default TCU resolution in accordance with the registry.<br>**Notice:** The first TCU panel to logon (later) becomes activated. The focus handler then automatically sets the resolution to this TCU panel's resolution (in the case of dynamic resolution switching).<br>A) in inactive TCU mode:<br>The resolution is not specially set - i.e. the resolution used during the previous session in the system is active, e.g. the resolution set manually in Control Panel. |
| AUTO_MON_2 | Like AUTO_MON_1, except:<br>Example 2: There is no PCU monitor, however there is a PCU panel (irrespective of whether there are any TCU panels):<br>[in active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is set to the max. resolution of the PCU panel (max. 1280x1024). |
| 640X480 | During boot up, the SINUMERIK resolution is set to 640x480. |
| 800X600 | During boot up, the SINUMERIK resolution is set to 800x600. |
| 1024X768 | During boot up, the SINUMERIK resolution is set to 1024x768. |
| 1280X1024 | During boot up, the SINUMERIK resolution is set to 1280x1024. |

# 1.4 BIOS settings for PCU 50.3

## Overview

The BIOS of the PCU is preset in such a way that no changes are required. The date and time can be set under Windows or the user interface.

**Note**

**Changes in the BIOS**

Your device configuration is preset for working with the software supplied with the unit.

You should only change the preset values if you have modified your device in any way, or if a fault occurs when the unit is powered up.

## Starting BIOS setup

1. Start the BIOS SETUP as follows:
   Reset the device (warm or cold restart).
   After the startup test, the following message appears:
   `Press < F2 > to enter SETUP or <ESC> to show boot menu`
2. Press the F2 key as long as the BIOS prompt appears on the screen.
   The BIOS main menu opens:

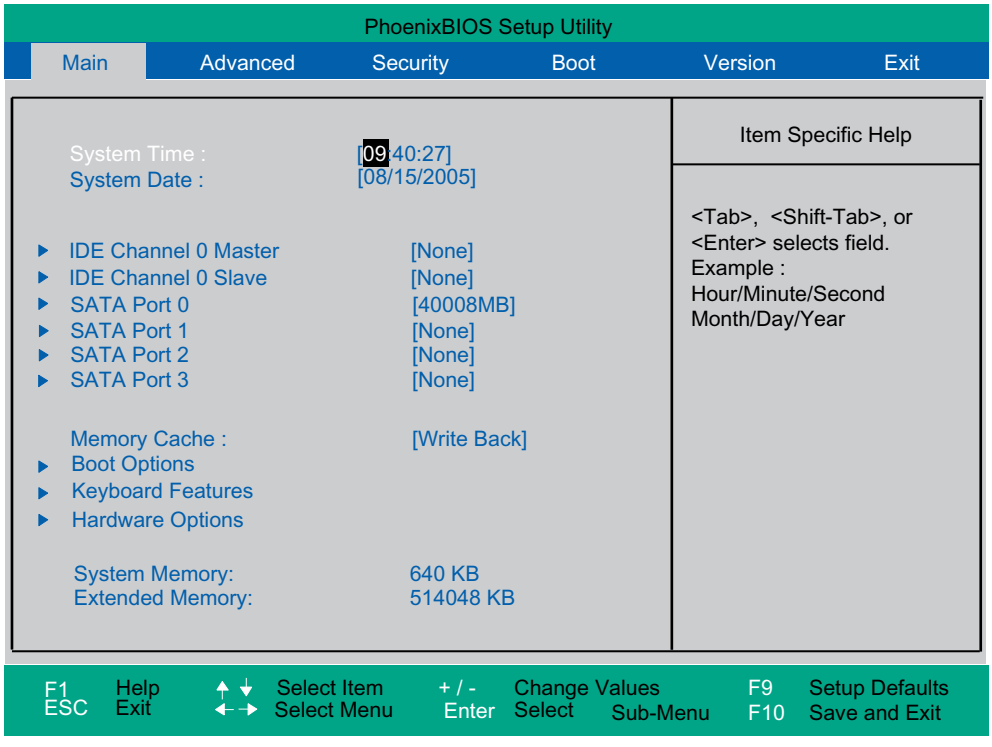



Figure 1-15 BIOS Main Menu (Example)

## BIOS setup: Default settings

The following system parameters are saved on delivery:

**Menu: Main**

| System parameters | Default settings | Own entries |
|---|---|---|
| System Time | hh:mm:ss | |
| System Date | MM/DD/YYYY | |
| IDE Channel 0 Master | None | |
| IDE Channel 0 Slave | None | |
| SATA Port 0 | 40008 MB | |
| SATA Port 1 | None | |
| SATA Port 2 | None | |
| SATA Port 3 | None | |
| Memory Cache | Write Back | |

| Boot options | | |
|---|---|---|
| Quick boot mode | Enabled | |
| SETUP prompt | Enabled | |
| POST errors | All, but not keyboard | |
| Summary screen | Enabled | |
| Diagnostic screen | Enabled | |
| Post Code/Status | LPC Bus | |

| Keyboard features | | |
|---|---|---|
| Numlock | ON | |
| Key click | Disabled | |
| Keyboard auto-repeat rate | 30 / sec | |
| Keyboard auto-repeat delay | ½ sec | |

| Hardware options | | |
|---|---|---|
| PCI MPI/DP | Enabled | |
| Onboard Ethernet 1 | Enabled | |
| On-board Ethernet 1 Address | 08 00 06 90 xx xx | |
| On-board Ethernet 1 Remote Boot | Enabled | |
| Onboard Ethernet 2 | Enabled | |
| On-board Ethernet 2 Address | 08 00 06 90 xx xx | |
| On-board Ethernet 2 Remote Boot | Disabled | |
| SafeCard functions | Enabled | |
| Fan control | Enabled | |
| CRT/LCD selection | Simultan. Auto | |

### Menu: Advanced

| System parameters | Default settings | Own entries |
|---|---|---|
| Installed O/S | Other | |
| Reset configuration data | No | |
| Legacy USB support | Disabled | |
| USB controller restart | Enabled | |
| I/O Device Configuration | | |
| Internal COM 1 | Enabled | |
| Base I/O address | 3F8 | |
| Interrupt | IRQ 4 | |

| PCI Configuration | | |
|---|---|---|
| **PCI device slot 1** | | |
| Option ROM scan | Enabled | |
| Enable master | Enabled | |
| Latency timer | Default | |
| **PCI device slot 2** | | |
| Option ROM scan | Enabled | |
| Enable master | Enabled | |
| Latency timer | Default | |

| SATA/PATA Configuration | | |
|---|---|---|
| PATA Controller: | Enabled | |
| SATA Controller mode | Enhanced | |
| AHCI Configuration | Disabled | |
| RAID support | Disabled | |

### Menu: Security

| System parameters | Default settings | Own entries |
|---|---|---|
| Supervisor password is | Disabled | |
| User password is | Disabled | |
| Set user password | Enter | |
| Set supervisor password | Enter | |
| Password on boot | Disabled | |
| Fixed disk boot sector | Standard | |

### Menu: Boot

| System parameters | Default settings | Own entries |
|---|---|---|
| Boot priority order: | | |
| 1: SATA0: | Fujitsu MHT2040BHTBD | |
| 2: PCI BEV: | VIA BootAgent | |
| 3: | | |
| 4: | | |
| 5: | | |
| 6: | | |
| 7: | | |
| 8: | | |
| Excluded from boot order: | | |

### Menu: Version

| System parameters | Default settings | |
|---|---|---|
| SIMATIC PC | SINUMERIK PCU 50.3 | |
| BIOS version | V05.01.06 | |
| BIOS number | A5E00370214-ES005 | |
| MPI/DP firmware | V01 | |
| CPU type | Celeron ® M processor 1.50 GHz | |
| CPU ID | 06D8 | |
| Code revision | 0020 | |

### Menu: Exit

| | |
|---|---|
| **Save Changes & Exit** | All changes are saved; a system restart is carried out with the new parameters. |

## Changing BIOS settings

Once additional components have been installed or attached, it may be the case that the system has to be informed of this via the BIOS setup:

1. Start the device.
2. When the prompt to activate the BIOS setup appears, press the <F2> key (corresponds to horizontal softkey 2 on the OP).
3. The BIOS setup menu appears. In the menu, use the cursor keys to navigate to the desired selection box.
4. Change the setting using the <+> key (press <SHIFT> and <X> at the same time) or the ↔ key on the numeric keypad.
5. Using the left-right cursor keys, you can reach other setup menus.

6. Press <ESC> (<Alarm Cancel> key) to go to the "Exit" menu (or press the right cursor key again).
7. Press the <Enter> key to exit the setup menu.

Then the system starts.

**Note**

Changes to the BIOS settings, with the exception of the boot sequence, require an OEM contract to be concluded.

# 1.5 BIOS settings for PCU 50.5

## Overview

The BIOS of the PCU is preset in such a way that no changes are required. The date and time can be set under Windows or the user interface.

**Note**

**Changes in the BIOS**

Your device configuration is preset for working with the software supplied with the unit.

You should only change the preset values if you have modified your device in any way, or if a fault occurs when the unit is powered up.

## Starting BIOS setup

1. Start the BIOS SETUP as follows:
   Reset the device (warm or cold restart).
   After the startup test, the following message appears:
   `PRESS < F2 > to enter SETUP or <ESC> to show Bootmenu`
2. Press the F2 key as long as the BIOS prompt appears on the screen.
   The BIOS main menu opens:

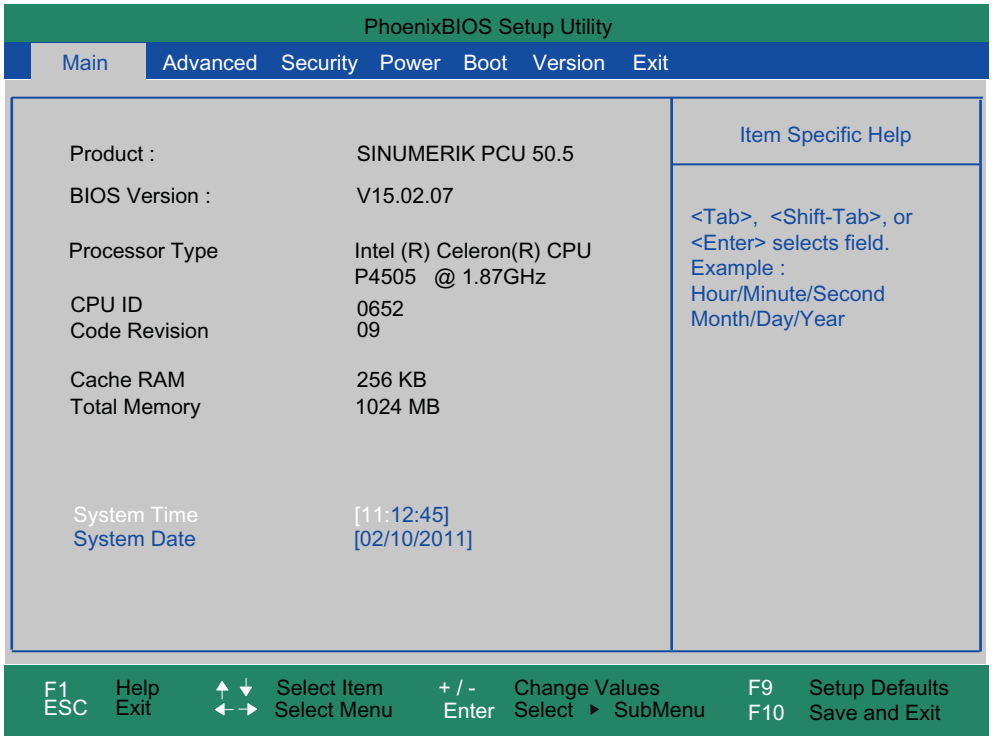



Figure 1-16 BIOS main menu PCU 50.5 (example)

## BIOS setup: Default settings

The following system parameters are saved on delivery:

### Menu: Main

| System parameters | Default settings | Own entries |
|---|---|---|
| System Time | hh:mm:ss | |
| System Date | MM/TT/JJJJ | |

### Menu: Advanced

| Peripheral Configuration | Default settings | Own entries |
|---|---|---|
| Internal COM 1 | Enabled | |
| Onboard Ethernet 1 | Enabled | |
| Onboard Ethernet 2 | Enabled | |
| PCI - MPI / DP [1)] | Enabled | |

[1)] only for CPB (-C)

| SATA Configuration | Default settings | Own entries |
|---|---|---|
| SATA Controller | Enabled | |
| SATA Controller mode | AHCI | |

| Video Configuration | Default settings | Own entries |
|---|---|---|
| Primary video device | PEG | |

| USB Configuration | Default settings | Own entries |
|---|---|---|
| USB Port0 | Enabled | |
| USB Port1 | Enabled | |
| USB Port2 | Enabled | |
| USB Port3 | Enabled | |
| USB Port4 | Enabled | |
| USB Port5 | Enabled | |
| USB Port6 | Enabled | |
| USB Port7 | Enabled | |
| USB Port8 | Enabled | |
| USB Port9 | Enabled | |
| USB Port10 | Enabled | |

| Chipset Configuration | Default settings | Own entries |
|---|---|---|
| Port 80h Cycles | LPC Bus | |
| VT-d | Enabled | |
| HPET | Enabled | |

| CPU Configuration | Default settings | Own entries |
|---|---|---|
| P-States (IST) | Enabled | |
| Turbo Mode | Enabled | |
| CMP Support | Auto | |
| HT Support | Auto | |
| Use XD Capability | Disabled | |
| VT Support | Enabled | |
| Turbo Mode | Disabled | |
| C-States | Enabled | |

| Active Management Technology Support | Default settings | Own entries |
|---|---|---|
| Intel AMT Support | Disabled | |
| Intel AMT Setup promt | Disabled | |
| AMT CIRA Request Trig | Disabled | |
| AMT DIRA Timerout | 30 | |
| Un-Configure ME | Disabled | |
| USB Configure | Enabled | |

| Advanced | Default settings | Own entries |
|---|---|---|
| Fan control | Enabled | |
| Operating Mode Switch | Enabled | |

### Menu: Security

| System parameters | Default settings | Own entries |
|---|---|---|
| Supervisor Password | Not installed | |
| User Password | Not installed | |
| Set Supervisor Password | Inactive (no password assigned) | |
| Set User Password | Inactive (no password assigned) | |

### Menu: Power

| System parameters | Default settings | Own entries |
|---|---|---|
| Wake on LAN 1 | Disabled | |
| Wake on PME/LAN2 | Enabled | |
| Wake on Time | Disabled | |
| After Power Failure | Power On | |

### Menu: Boot

| System parameters | Default settings | Own entries |
|---|---|---|
| Quick Boot | Disabled | |
| Quiet Boot | Disabled | |
| POST Errors | All without keyboard | |
| NumLock | on | |
| USB Boot | Enabled | |
| Bootmanager | Enabled | |
| PXE Boot to LAN 1 | Enabled | |
| PXE Boot to LAN 2 | Disabled | |

| Legacy | Default settings | Own entries |
|---|---|---|
| Normal Boot Menu | Advanced Placeholder | |

### Menu: Version

| System parameters | Default settings |
|---|---|
| Product | SINUMERIK PCU 50.5 |
| BIOS Version | V15.02.07 |
| BIOS Number | A5E02619467-ES001 |
| InsydeH20 Version | 03.59.53.1053 |
| Intel ME Version | 6.0.3.1203 |
| Video Option ROM | 1976 |
| RAID Option ROM | 9.5.0.1037 |
| PXE Option ROM | 1.3.30 |
| Mode switch 0 | Normal |

### Menu: Exit

| System parameters | Default settings | Own entries |
|---|---|---|
| **Exit Saving Changes** | All changes are saved; a system restart is carried out with the new parameters. | |
| Profile | Standard | |

## Changing BIOS settings

Once additional components have been installed or attached, it may be the case that the system has to be informed of this via the BIOS setup:

1. Start the device.
2. When the prompt to activate the BIOS setup appears, press the <F2> key (corresponds to horizontal softkey 2 on the OP).
3. The BIOS setup menu appears. In the menu, use the cursor keys to navigate to the desired selection box.
4. Change the setting using the <+> key (press <SHIFT> and <X> at the same time) or the ↔ key on the numeric keypad.
5. Using the left-right cursor keys, you can reach other setup menus.
6. Press <ESC> (<Alarm Cancel> key) to go to the "Exit" menu (or press the right cursor key again).
7. Press the <Enter> key to exit the setup menu.

Then the system starts.

**Note**

Changes to the BIOS settings, with the exception of the boot sequence, require an OEM contract to be concluded.

## Boot sequence

### BIOS version 05.01.12

In order to ensure correct functioning of the Software Ghost with Backup/Restore, please set in BIOS under "SATA/PATA Configuration" → "SATA Controller Mode" → "Compatible".

The CompactFlash Card is entered as SATA. The hard disk (or Solid State Drive) is DISK0, and the CompactFlash Card is DISK2.

# Configuring the system 2

## 2.1 System settings

### 2.1.1 Configuring the communication via MPI

If you are using a PCU in conjunction with a SINUMERIK 840D *powerline* and require MPI communication, you must install the S7DOS 8.4 communication driver. In all other cases, you do not have to install this driver.

The setup for this communication driver can be found in the following directory:

- `D:\updates\`

### 2.1.2 How to change the name of the PCU

#### Default

The PCU is supplied with an automatically generated computer name.

### Procedure

To change the name of the PCU:

1. Select "Start" → "Control Panel" → "System".
2. Select the "Computer Name" tab and click on "Change".
   The following dialog opens:

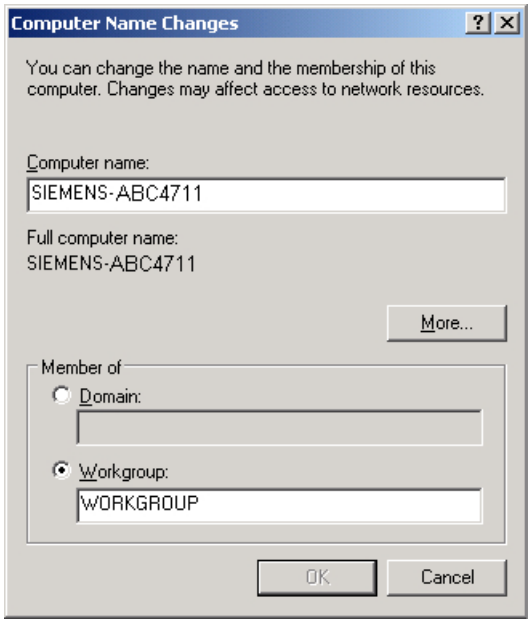



Figure 2-1 Changing the name of the PCU

## 2.1.3 This is how you set the IP address of the PCU

### Default setting

**Note**

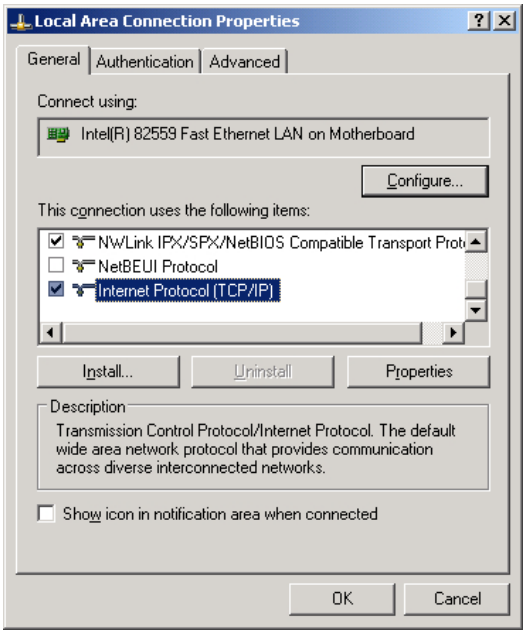
**IP address in the system network**

The IP address 192.168.214.241 is set as a factory default for every PCU on the system network.

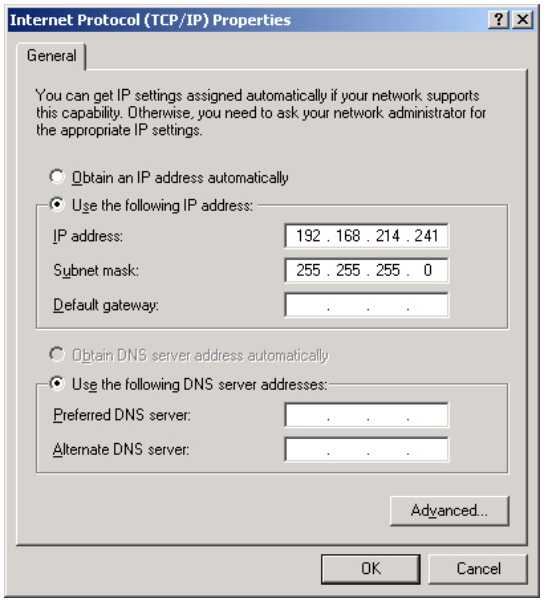
You must only perform the steps described below if you wish to change this default.

## Procedure

1. Select the following on the PCU on the Service Desktop: "Start" → "Settings" → "Network Connections".
   The "Network Connections" window opens.
2. Double-click the Ethernet 2 interface you want to parameterize which is to be used for connecting the TCU or system network.
   The "Ethernet 2 (System Network) Properties" window opens.
3. Under the "General" tab, select "Internet Protocol (TCP/IP)" and click the "Properties" button.
   The "Internet Protocol (TCP/IP) Properties" window opens:



4. Under the "General" tab, select the "Use the following IP address" option and enter the IP address and the subnet screen form.
   Recommended setting for the first PCU:




5. Enter the required new IP address and confirm the settings with "OK".

## 2.1.4 How to add the PCU to a domain

### Requirement

Only a user with the corresponding entitlement, e.g. a domain administrator, can add a PCU to an existing domain.

### Domain Controller (DC)

A domain controller (DC) is a server for central authentication and authorization of computers and users in a network. In a network with a domain controller, several computers are combined to form one domain.

### Proceed as follows

To add the PCU to a domain:

1. Select: "Start" → "Settings" → "Control Panel" → "System", "Computer Name" tab.
   Default: The PCU belongs to a "WORKGROUP" and is not assigned to any domain.
2. Click "Change ...".
   The following dialog opens:

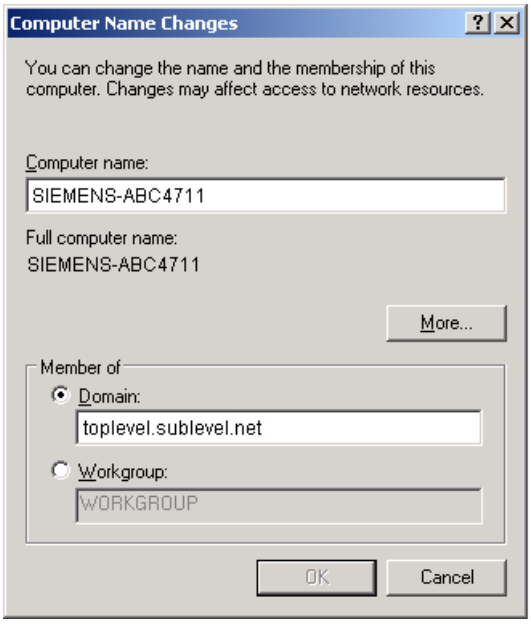



Figure 2-2 Add PCU to domain

3. Enter the name of the domain to which you want to add the PCU.
4. You will then be asked to log on as a user with the corresponding entitlement to conclude the process.

## 2.1.5 How to connect an external monitor

### Preconditions

To connect an external monitor, the following preconditions apply:

- The external monitor is connected to the DVI interface on the PCU (using an adapter, if necessary).
- The monitor may not be connected while in use.

### Procedure

To connect an external monitor:

1. Right-click the PCU's service desktop and select "Properties" from the pop-up menu.
2. Select the "Settings" tab and then click "Advanced".

3. Select the "Troubleshooting" tab and and set the "Hardware acceleration" to a value other than zero. The recommended setting is "full".
4. Close the dialog and click OK to confirm all the dialogs.
5. Repeat steps 2 and 3. There is an additional tab for "Intel(R) ... Grafics Controller"
6. Click the "Graphics Properties" button.
   - In the case of a PCU 50.3: On the "Devices" tab the external monitor corresponds to the "Monitor" selection; the OP/TP on the PCU corresponds to the "Notebook" selection. Select a "Primary Device" and a "Secondary Device".
   - In the case of a PCU 50.5: On the "Display" tab, the OP/TP on the PCU corresponds to the selection "Primary Display"; the external monitor corresponds to the selection "Secondary Display". Click the button "Multiple Displays" to select the "Clone" or "Extended" mode.
7. Close the dialog and click OK to confirm all the dialogs: The external monitor is now ready.

## 2.1.6 This is how you switch off the USB interfaces.

### Command sc_usb disable

To prevent harmful software entering via the USB ports to the control or into the system network, you can switch off the USB interfaces.

This command is called up in the DOS shell:

| | |
|---|---|
| Default setting: | The USB interfaces are not locked. |
| Syntax: | sc_usb disable [-minutes] [all] [HOSTS...] |

- Without a time indication [-minutes], a permanent disable is set. With time indication, the disable takes the time set in minutes.
- The disable may refer to all network nodes of the system network, or a list of host names or IP addresses may be specified to which the disable should apply.

### Command sc_usb enable

With "sc_usb enable" USB storage units on permanently disabled USB interfaces or on certain host names or IP addresses are enabled again.

# 2.2 Configuring a customized operator interface

## 2.2.1 How to select the language for the Windows system

### Default setting

In the delivery condition, the Windows XP operating system is only installed on the PCU in English and with a US keyboard layout.

### Requirement

In order to be able to switch languages, the desired languages must be installed from the DVD of the "SINUMERIK Service Pack Recovery Media Win XP ProEmbSys SP2". With the "Multilingual User Interface" (MUI), you can switch to menus, dialog boxes and keyboard layouts for the Windows system in different languages:

**⚠ WARNING**

**"Regional Options" tab**

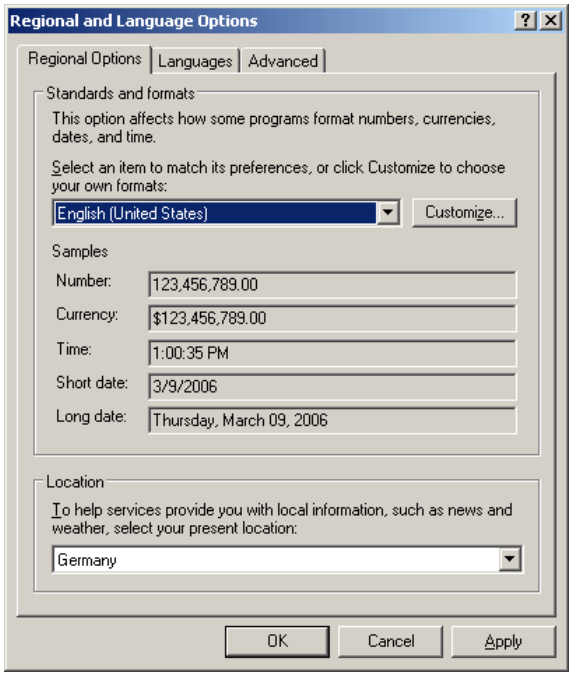
The settings for the keyboard assignment and the formats for date, time and number displays on the "Regional Options" tab **must not be changed under "Standards and formats"**.

These settings are automatically adapted depending on the language selected for the operating software.

## Select a language

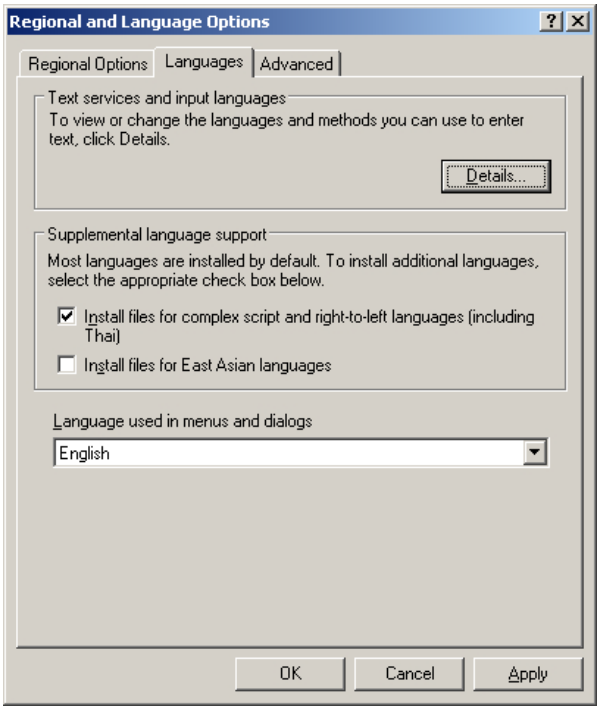
After installing a language from the corresponding CD, proceed as follows:

1. Select "Start" → "Control Panel" → "Language and Regional Options", to open the following dialog box:




This setting must not be changed!

2. Select the "Languages" tab, in order to switch the language for the Windows XP user interface. Under "Language used in menus and dialogs," select the new language and confirm with OK.




3. On the "Advanced" tab, select the language for programs that do not support Unicode.

### Result

To make the language change effective, the PCU must be rebooted. The selectable languages are displayed using the font set of the respective language.

### See also

How to install additional languages under Windows XP (DVD) (Page 74)

## 2.2.2 Displaying the boot screen

### Default setting

The Siemens boot screen is archived in the PCU under the following path:

`F:\hmisl\siemens\sinumerik\hmi\ico\ico640\splash.png`

### Display OEM-specific boot screen

You can choose any name.

The directory is set in ServiceCenter Users under:

- Startup: Bitmaps folder
- Startup: Default bitmap folder

## 2.2.3 Changing the background of the service desktop

You can configure user-specific background pictures for the Service Desktop in the ServiceCenter Users.

If you want to set a background pattern instead, make this setting in the registry.

### Procedure

#### Background picture

1. Save the required pictures in the appropriate resolutions in the corresponding subfolders of `F:\Background picture`
2. Switch to the appropriate "User Types" in the ServiceCenter Users and specify the `F:\Background picture` folder at "Background" in the "Desktop" tab.

**Note**

An attempt to set the Service Desktop background (as in standard Windows) via the Control Panel does not affect the Service Desktop background display, but only the startup screen display.

#### Background pattern

A background pattern for the Service Desktop is not set via the "Control Panel" (system control), but in the registry:

| | |
|---|---|
| Key: | HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\ <version>\HMI-Desktop |
| Value: | Pattern (STRING) |
| Date: | <bit code>" (e.g. "0 80 114 32 0 5 39 2",<br>see HKCU\Control Panel\Patterns) |
| | "(None)" (= NO background pattern) |
| InitDatum: | Background pattern previously set via the Control Panel |
| Default data: | "(None)" (if entry is not available/readable) |

# 2.3 Customized settings during boot up

## Overview

Regarding the settings and functionality, booting followed by the start of an HMI program starting differs from booting with subsequent start of the Service Desktop.

## 2.3.1 Configuring key filters for an HMI program

### Filtering keys

During boot up of an HMI program, keystroke sequences and pressed function keys are simultaneously filtered. The keystroke sequences and functions that were pressed simultaneously and are to be filtered are configured in file E:\Windows\System.ini.

Filtering keystroke sequences:

| | |
|---|---|
| Section: | MMC103Keyb |
| Key: | SeqAct |
| Value: | <bit mask><br>(= keystroke sequences to be filtered, specified in accordance with the comment in E:\Windows\System.ini) |
| Init value: | 262143 |

Filtering function keys that were pressed simultaneously:

| | |
|---|---|
| Section: | MMC103Keyb |
| Key: | ConcurrentKeyMask |
| Value: | <bit mask><br>(= function keys to be filtered, specified in accordance with the comment in E:\Windows\System.ini) |
| Init value: | 255 |

In the ServiceCenter Users, the keys that are to be filtered can be configured for each specific user:

- while the system boots → "Startup (2)" tab
- while the HMI program runs: → "HMI" tab
- on the Service Desktop or Windows Desktop: → "Desktop (2)" tab

## system.ini file

```
# ----------------------------------------
# SIEMENS KEYBOARD-DRIVER SECTION (START)
# ----------------------------------------
# State of driver after startup
# (0=NORMAL, 1=PERMANENT-SHIFT ACTIVE)
keybStartState=0

# Type of Keyboard to be used
# (0=MMC103, 1=MFII)
keybType=0

# Filtering Key Sequences
# =======================
# Activation of Key-Sequences to be ignored
# SeqAct holds a BIT-Pattern for a max of 20 sequences
# which could be ignored by the keyboard-driver
# if bit-n is set in Seq-Act, the according sequence
# will be ignored
# The following 20 sequences are implemented
# 0  CTRL-ALT-DEL
# 1  ALT-F4
# 2  ALT-TAB
# 3  LEFTSHIFT-ALT-TAB
# 4  RIGHTSHIFT-ALT-TAB
# 5  CTRL-ESC
# 6  ALT-ESC
# 7  ALT-SPACE
# 8  (SHIFT)-CTRL-F1
# 9  (SHIFT)-CTRL-F2
# 10 (SHIFT)-CTRL-F3
# 11 (SHIFT)-CTRL-F4
# 12 (SHIFT)-CTRL-F5
# 13 (SHIFT)-CTRL-F6
# 14 (SHIFT)-CTRL-F7
# 15 (SHIFT)-CTRL-F8
# 16 M$_1
# 17 M$_2
# 18 CAPSLOCK
# 19 NUMLOCK
# 20 (reserved)
KeySequencesEnable =1
SeqAct=262143
```

```
# Filtering Concurrent Keys
# ========================
# Definition and selection of function keys (buttons) which shall not be
# pushed in parallel or at the same time (function keys F1-F12 and
# additional hardkeys).
# If more of these buttons are pushed at the same time, the system removes the
# code of these buttons except the code of the button which was pushed first.
# This function of the keyboard-driver is activated by ConcurrentKeyEnable.
# ConcurrentKeyMask holds a BIT-Pattern for a max of 17 function keys, which
# must not be pushed simultaneously (concurrently). If bit-n is set in
# ConcurrentKeyMask, the corresponding button is supervised.
# The following 17 function keys are implemented:
# F1
# F2
# F3
# F4
# F5
# F6
# F7
# F8
# F9
# F10
# F11
# F12
# PAGE-UP     (NUMLOCK) Alarm Hardkey on Operator Panel
# PAGE-DOWN   (NUMLOCK) Tool Management Hardkey on Operator Panel
# HOME        (NUMLOCK) Program Manager Hardkey on Operator Panel
# END         (NUMLOCK) Program Hardkey on Operator Panel
# CURSOR-DOWN (NUMLOCK) Alarm Hardkey on Operator Panel
```

**Note**

**Scope of the keys**

Only keys, which have already been defined in the system.ini file, can be enabled.

No additional keys can be defined for filtering.

## See also

ServiceCenter Users: User settings (Page 10)

## 2.3.2 Saving service desktop settings

### Starting the service desktop

The Windows platform is freely accessible via the service desktop. When the service desktop is started, all of the programs that would automatically be started by Windows (standard version) during log-in are also started.

- **Executing an HMI program**
  The HMI program can also be started from the service desktop.
- **Ending an HMI program**
  When an HMI program started from the service desktop is exited, you are returned to the service desktop.

### Saving the service desktop (default)

The settings on the service desktop (e.g. arrangement of the links on the service desktop) are not saved when you log off. A service technician should always find the same starting condition on the service desktop, not the settings from a previous session.

### Saving the settings of the service Desktop

This behavior can be changed by making an entry in the registry. The following settings can be saved via this registry entry:

- Positions of open windows
- Size and position of the task bar
- Moving and deleting links

| | |
|---|---|
| Key: | HKLM\SOFTWARE\Siemens\SINUMERIK\Basic software\ <version>\HMI Manager |
| Value: | SaveSINDesktopSettings (DWORD) |
| Date: | 1 (the settings are saved) or<br>0 (the settings are NOT saved) |
| Init data: | - Value is not created by the basic software - |
| Default data: | 0 (if entry is not available/readable) |

The key is effective for all service users and other users.

Links on the service desktop are always saved, irrespective of the registry entry.

**Note**

Application windows that are still open before logging out, must be closed by the setting "Save settings" before exiting the service desktop. Otherwise, these application windows will briefly be displayed and then closed again during a restart immediately before the HMI program starts.

## 2.3.3 Starting programs during boot up

### Starting additional programs

Programs can also be started at the same time as the HMI program and are started automatically by Windows when the service desktop is opened. This start in parallel to the HMI program can be configured.

If the programs to be started are located in the Windows directories of E:\Documents and Settings, the following registry entry must be set:

| | |
|---|---|
| Key: | HKLM\SOFTWARE\Siemens\SINUMERIK\Basic software\ <version>\HMI Manager |
| Value: | StartSINHMIStartupDirsPrograms (DWORD) |
| Date: | 1 (the programs are started) or<br>0 (the programs are NOT started) |
| Init data: | 0 |
| Default data: | 0 (if entry is not available/readable) |

If the programs to be started are set in the registry entries 'HKCU\Software\Microsoft\Windows \CurrentVersion\Run' and 'HKLM\Software\Microsoft\Windows\CurrentVersion\Run', the following registry entry must be set:

| | |
|---|---|
| Key: | HKLM\SOFTWARE\Siemens\SINUMERIK\Basic software\ <version>\HMI Manager |
| Value: | StartSINHMIRunPrograms (DWORD) |
| Date: | 1 (the programs are started) or<br>0 (the programs are NOT started) |
| Init data: | 0 |
| Default data: | 0 (if entry is not available/readable) |

### Executing an HMI program

While the HMI program is being executed, Windows Explorer runs in the background and its settings (disabled browser functionality, taskbar, start menu, ...) prevent the Windows platform being accessed unintentionally.

The behavior of the taskbar can be user-specifically set in the ServiceCenter Users under "HMI program" using "Taskbar Autohide" and "Taskbar On Top".

### Ending an HMI program

When the HMI program is ended, Windows XP shuts down completely.

## 2.3.4 Starting OEM programs

### Overview

You can start OEM programs directly before starting the HMI system software. This requires these programs (or their links) to be stored in subdirectories of the directory C:\RunOEM.

### Starting sequence

The subdirectories are executed in the order listed. The programs within a subdirectory are started in the chronological order in which they were placed in the subdirectory.

- Programs in the C:\RunOEM\SeqOnce subdirectory are started **once and sequentially**, i.e., a program is not started until the previously started program is completed.
- Programs in the C:\RunOEM\Seq subdirectory are started **sequentially whenever the system is ramped up**, i.e., a program is not started until the previously started program is completed.
- Programs in the C:\RunOEM\ParOnce subdirectory are started **once and simultaneously**. They run parallel with the HMI system software.
- Programs in the C:\RunOEM\Par subdirectory are started **simultaneously whenever the system is ramped up**. They run parallel with the HMI system software.

Not only program files, but also other types of file can be stored in the subdirectories, which are then opened in accordance with their file type.

For example, ".txt" files are opened using Notepad, ".htm" files are opened using Internet Explorer.

## 2.3.5 Starting applications in service mode

### Starting other applications

If other applications are to be started in service mode, enter them with their complete path in the [OEMRun] section in the file WINBOM.INI:

**Example: Starting the "Notepad" program**

```
[OEMRunOnce]
"Start WinVnc", "x:\I386\system32\StartWinVnc.exe"
"Check Password","x:\I386\system32\CheckPEPwd.exe"
[OEMRun]
"Start Backup/Restore", "x:\I386\system32\GhostOrder.exe"
"notepad","e:\windows\notepad.exe"
```

All other entries must not be changed.

# 2.4 Configuring the SITOP UPS module for use with the PCU Base Software

## Overview

SITOP UPS modules can continue operation of the PCU 50.5 temporarily if a power failure occurs and/or shut down the system properly.

For example, a SITOP UPS module also protects against data loss when the PCU 50.5 hardware is shut down, because for technical reasons with Windows-based systems, data is still written to the SSD after shutdown.

- You will find general information about SITOP UPS modules used in combination with the PCU base software in Section SITOP modules for the PCU 50.5 (Page 56).
- If you want to use a SITOP UPS module in combination with the PCU base software, you must first adapt the settings of the SITOP software (Page 57).
- Optionally, you can adapt the delay time after which the HMI software will be forced to shut down (Page 60).
- You must configure the hardware (Page 62) of the SITOP UPS module before using it with the PCU base software.

## 2.4.1 SITOP modules for the PCU 50.5

### Suitable SITOP UPS modules

SITOP UPS modules can continue operation of the PCU 50.5 temporarily if a power failure occurs and/or shut down the system properly.

In conjunction with the PCU base software, all specific applications are taken into account (e.g. HMI software) with the `Shutdown.bat` component.

Safe shutdown after a power failure of the delivery condition of the PCU 50.5 takes approximately 20 seconds. For a typical power consumption of approx. 60 W, a capacitor-buffered UPS with an energy storage device of 2.5 kW is recommended for a PCU 50.5. If longer buffer times are required, then several expansion modules can be connected in a cascade connection.

**Example of a compatible SITOP UPS module:**

SITOP 500S (15 A / 2.5 kWs capacitor)

Article no.: 6EP1933-2EC41

### Further information

Information about available SITOP UPS modules is given in the catalog or in the Internet:

- Catalog KT 10.1 Power supply SITOP
- SIEMENS Industry Mall > 24 V DC Uninterruptible Power Supplies (https://eb.automation.siemens.com/mall/en/US/Catalog/Products/7010117)

**Note**

**Information about the test environment for machine OEMs**

The functionality has been tested in the standard configuration with the PCU base software.

When installing add-on or OEM software components, you must check the shutdown procedure of the entire system and adapt the factory setting (180 seconds).

See: Parameterizing a delay time for quitting the HMI software (Page 60)

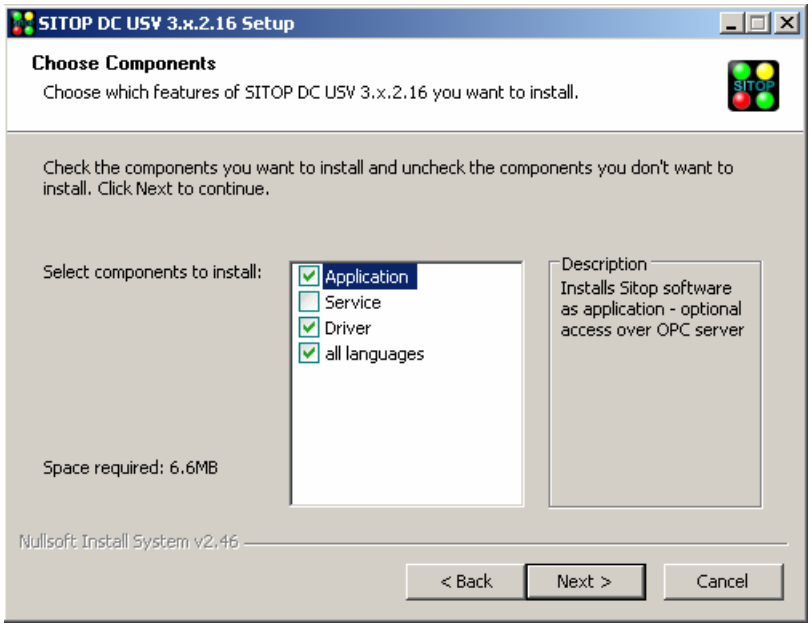
## 2.4.2 Configuring SITOP software for PCU Base Software

If you want to use a SITOP UPS module in combination with the PCU base software, you must first adapt the settings of the SITOP software.

### Requirement

- The SITOP UPS hardware is connected via the USB interface.
- The SITOP software of version 3.x.2.16 or later is installed in directory `C:\Program Files (x86)\SITOP\`

- If you want to use SITOP software of version 3.2.1.16 or older, this must be installed as a normal application, not as a service.




**Note**

**Windows service of the SITOP software version 3.2.1.16 or older is not compatible with the PCU base software**

If you use SITOP software of version 3.2.1.16 or older as a Windows service with the PCU base software, the correct shutdown procedure cannot be performed!

If you have installed the SITOP software of version 3.2.1.16 as a Windows service, you must uninstall this and install it again as an application in order to be able to use it in conjunction with the PCU base software.

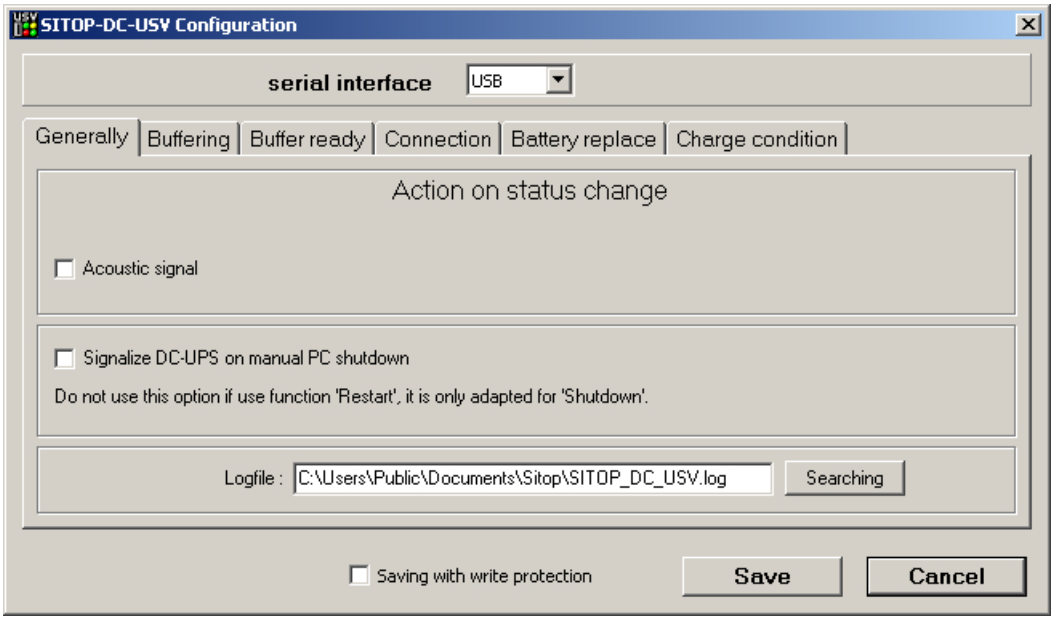
As of version 3.2.1.17, the SITOP software can be used as service with the PCU base software and is recommended.

- The SITOP software was configured during the installation so that it starts automatically when Windows is run-up (factory setting).
- The Service Desktop is active.

## Procedure

Proceed as follows to configure the SITOP software for use with the PCU base software:

1. Call the settings of the SITOP software, for example, by right-clicking the SITOP UPS icon in the Windows information area and then selecting "Configuration".
   The "SITOP DC UPS Configuration" dialog box opens.




2. In the "General" tab, select "UPS" in the drop-down list at "Serial interface".

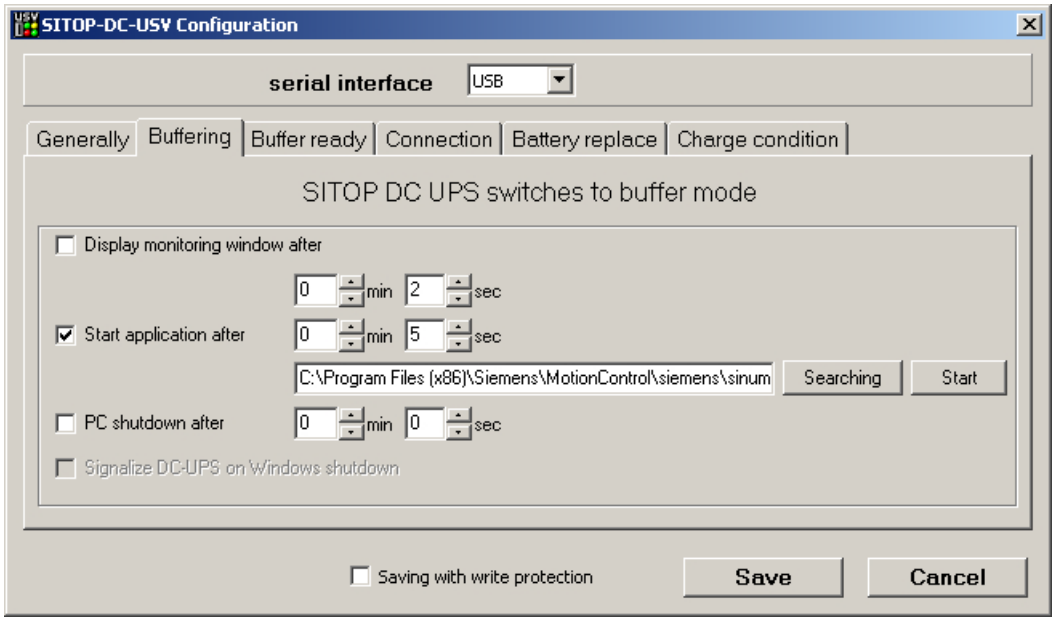
3. Switch to the "Floating operation" tab and make the following settings:
   - Deactivate the "Display monitoring window after" checkbox

     **Note**
     **Deactivating the monitoring window**
     Display of the monitoring window can cause malfunctioning of the HMI software.

   - Click directly in the text field at "Start application after" and specify the path of the `Shutdown.bat`:
     `C:\Program Files (x86)\Siemens\MotionControl\siemens\sinumerik\hmi\base\Shutdown.bat`

     **Note**
     **Do not use the "Browse" button**
     The dialog can crash if you use the "Browse" button to specify the path of the `Shutdown.bat`.
     Instead, enter the path directly in the text field.

   - After you have specified the path of the `Shutdown.bat`, activate the "Start application after" checkbox. Optionally, you can specify the wait time after which the `Shutdown.bat` is to be executed (e.g. 5 seconds).




4. Confirm the settings with "Save."

## 2.4.3 Parameterizing a delay time for quitting the HMI software

Before the SITOP monitor closes Windows 7, `Shutdown.bat` shuts down operation of the HMI software.

If the HMI software cannot be shut down within 180 seconds (factory setting) due to an error, HMI software shutdown is forced and Windows 7 is shut down.

If the HMI software of your OEM installation is not shut down within 180 seconds, you can parameterize this wait time manually.

**Note**

**Information about the test environment for machine OEMs**

The "SITOP UPS" function has been tested in the standard configuration with the PCU base software. When installing add-on or OEM software components, you must check the shutdown procedure of the entire system and adapt the factory setting.

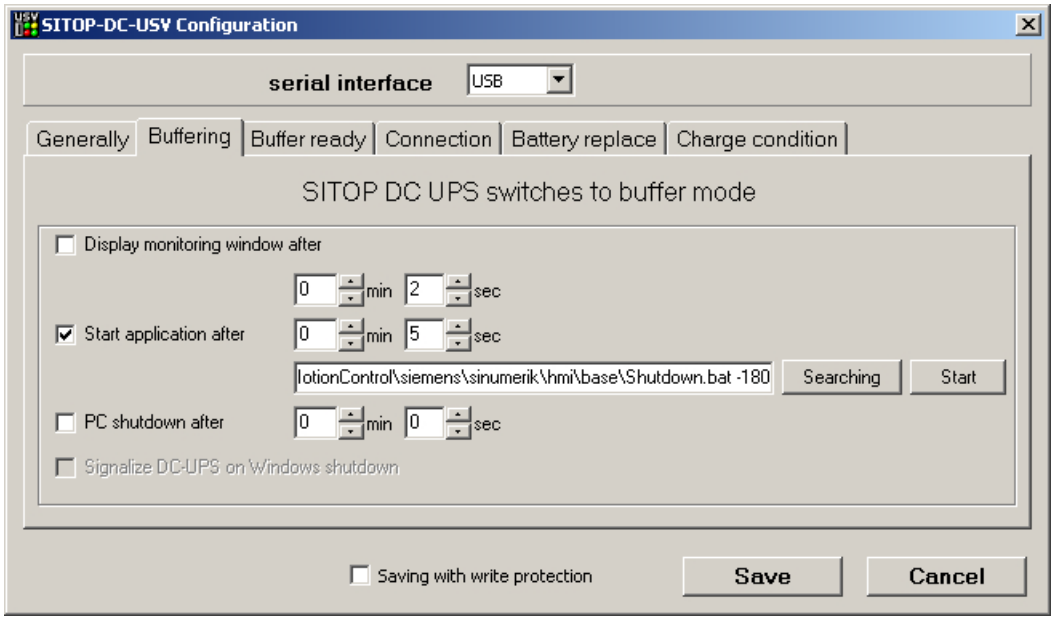
## Requirement

- The SITOP UPS module is configured for use with the base software.
  See Configuring SITOP software for PCU Base Software (Page 57).
- The Service Desktop is active.

### Procedure

Proceed as follows to change the delay time:

1. Call the settings of the SITOP software, for example, by right-clicking the SITOP UPS icon in the information area and then selecting "Configuration".
2. In the "SITOP DC UPS Configuration" dialog box, enter the wait time in seconds as a command line parameter in the "Floating operation" tab:
   - Syntax: `<Path>\Shutdown.bat -<Time in seconds>`
   - Example: `C:\Program Files (x86)\Siemens\MotionControl\siemens\sinumerik\hmi\base\Shutdown.bat -180`




3. Confirm the setting with "Save".

## 2.4.4 Hardware configuration of the SITOP UPS module

### Buffering parameterization

The UPS module can be used to select whether buffering should be completed after a predetermined period of time or not until the accumulator's lowest discharge threshold (= maximum buffer time) has been reached. Both buffering parameterizations result from this.

### "Maximum buffer time" mode

This mode enables the system to be shut down in a time-optimized manner. The UPS module is synchronized with the shutdown of the operating system. Buffering is maintained until the operating system has been shut down. The operating system must shut down within a maximum of five minutes (including all applications). Otherwise, the UPS module buffers for the maximum buffer time (dependent on the accumulator state).

**Required settings on the UPS module (USB interface)**

| | On - Off | | | |
|---|---|---|---|---|
| 1 | | ○ | +2V | Cut-in threshold<br>+22V fixed |
| 2 | | ○ | +1V | |
| 3 | ○ | | +0.5V | |
| 4 | | ○ | +1V | End-of-charge voltage<br>+ 26.3V fixed |
| 5 | | ○ | +1V | |
| 6 | ○ | | +0.5V | |
| 7 | ○ | | +0.2V | |
| 8 | | ○ | +0.2V | |
| 9 | | ○ | +0.1V | |
| 10 | | ○ | 0.35A / 0.7A | Charging current |

| | On - Off | | | |
|---|---|---|---|---|
| 1 | | ● | | Set time/max. time |
| 2 | | ○ | +320 s | Buffer time<br>+5 s fixed |
| 3 | | ○ | +160s | |
| 4 | | ○ | +80s | |
| 5 | | ○ | +40s | |
| 6 | | ○ | +20s | |
| 7 | | ○ | +10s | |
| 8 | ● | | | Disconnection |
| 9 | | ○ | | Battery operating state on/off |

Legend:

○ Delivery condition setting

● Setting for operation on the PCU

## "Fixed buffer time" mode

In this mode, the UPS module always buffers for the pre-selected, fixed period of time. It is not possible to synchronize the UPS module with the operating system shutdown.

### Required settings on the UPS module

| | On - Off | | | |
|---|---|---|---|---|
| 1 | | ○ | +2V | Cut-in threshold<br>+22V fixed |
| 2 | | ○ | +1V | |
| 3 | ○ | | +0.5V | |
| 4 | | ○ | +1V | ◦<br>◦<br>End-of-charge voltage<br>+ 26.3V fixed |
| 5 | | ○ | +1V | |
| 6 | ○ | | +0.5V | |
| 7 | ○ | | +0.2V | |
| 8 | | ○ | +0.2V | |
| 9 | | ○ | +0.1V | |
| 10 | | ○ | 0.35A / 0.7A | Charging current |

| | On - Off | | | |
|---|---|---|---|---|
| 1 | ● | | | Set time/max. time |
| 2 | | ○ | +320 s | ◦<br>◦<br>Buffer time<br>+5 s fixed |
| 3 | ● | | +160s | |
| 4 | | ○ | +80s | |
| 5 | | ○ | +40s | |
| 6 | | ○ | +20s | |
| 7 | | ○ | +10s | |
| 8 | ● | | | Disconnection |
| 9 | | ○ | | Battery operating state on/off |

Legend:

○ Delivery condition setting

● Setting for operation on the PCU

# 2.5 Configuration of the Service Center

## 2.5.1 Overview

You can permanently adapt the settings for the backup and restoration of disk images (Service Center (Page 77) component) in the "Service Center Backup/Restore" dialog box:

- Configuration of the network adapter (Page 66)
- Configuration of the host (Page 67)

Adaptations that you make here are stored in the `servicesystem.ini` configuration file.

You can also adapt the settings directly in the associated configuration files:

- `servicesystem.ini`
- Additional settings for the Service System on the USB flash drive: `ghost.ini`

You can also view the log files of the Service Center or restart the PCU 50.5 in the Service Center in the "Service Center Backup/Restore" dialog box.

To open the dialog box, click the ServiceCenter Backup-Restore icon on the Desktop.

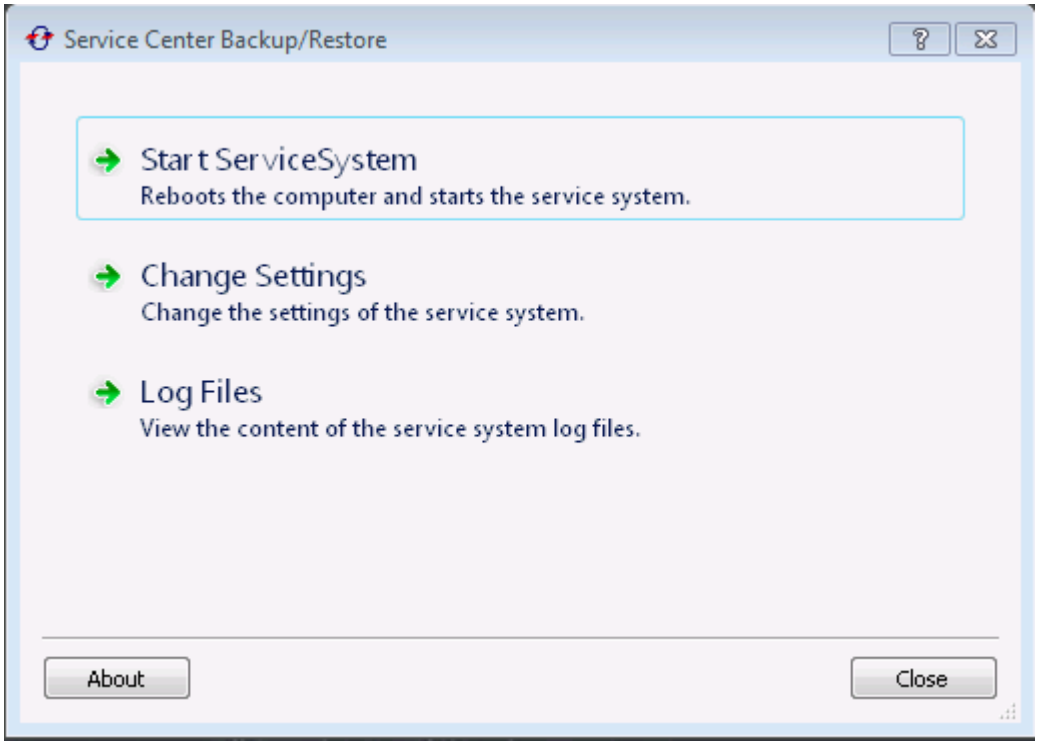



Figure 2-3 ServiceCenter Backup/Restore

## 2.5.2 Configuration of the network adapter

You can make the network settings in the "Adapter Settings" tab and permanently save this configuration, in contrast to the settings in the "Network Settings" dialog box of the Service Center (Page 81).

Settings that you make in this dialog box are stored in the `servicesystem.ini` configuration file.

### Overview

You can make the following IP settings in the "Adapter Settings" dialog box:

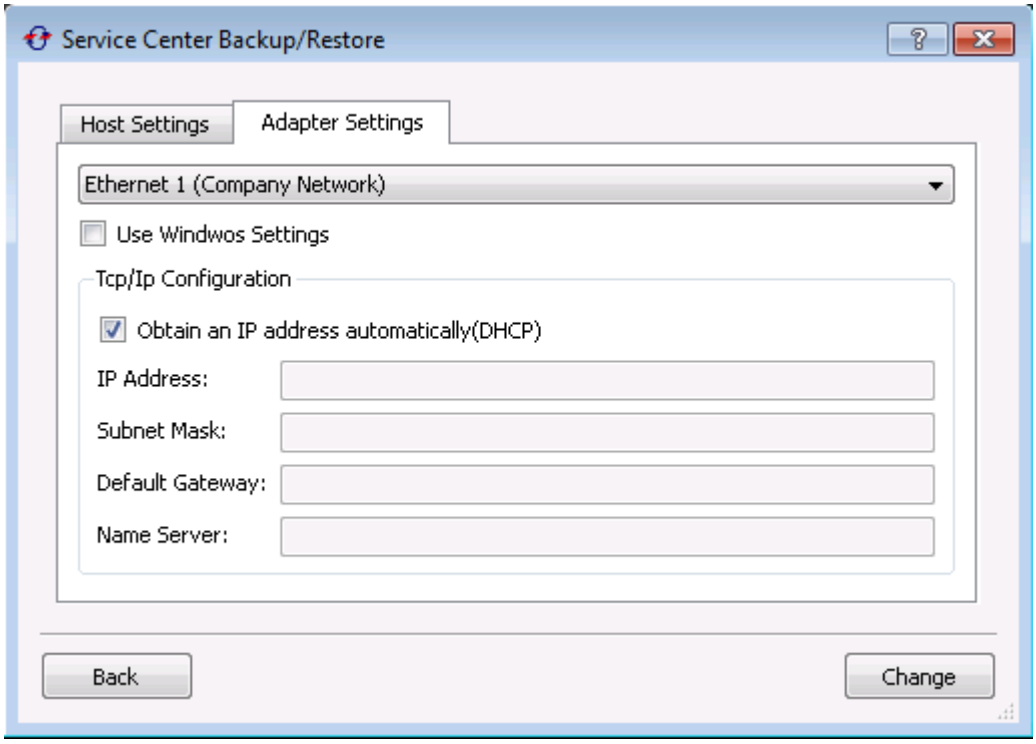



Figure 2-4 Adapter settings - Ethernet 1 (company network)

Table 2-1 Settings in the "Adapter Settings" dialog box

| Settings | Purpose |
|---|---|
| Adapter | Select the Ethernet interface that you want to configure.<br>All settings in this dialog box refer to the selected Ethernet interface. |
| Use Windows Settings | Select whether the values set in the Control Panel are to be used. |
| Obtain an IP address automatically (DHCP) | Select whether the addresses are to be fetched dynamically from the DHCP server.<br>If the checkbox is deactivated, you must make the settings manually. |
| IP Address | Set the IP address of the NCU.<br>You can specify an IP address from the following range:<br>192.168.214.250 - 254 |
| Subnet Mask | Specify a subnet mask, e.g. 255.255.255.0 |

| Settings | Purpose |
|---|---|
| Default Gateway | Set the IP address of the standard gateway.<br>A standard gateway creates a standard route in the IP routing table for all destinations that are not in the subnet. |
| Name Server | Set the IP address of the name server.<br>A name server answers questions asked about a domain name zone using a DNS database. |

## 2.5.3 Configuration of the host

You can make the network settings in the "Host Settings" tab and permanently save this configuration, in contrast to the settings in the "Network Settings" dialog box of the Service Center.

Settings that you make in this dialog box are stored in the `servicesystem.ini` configuration file.

### Overview

You can make the following settings in the "Network Settings" dialog box:

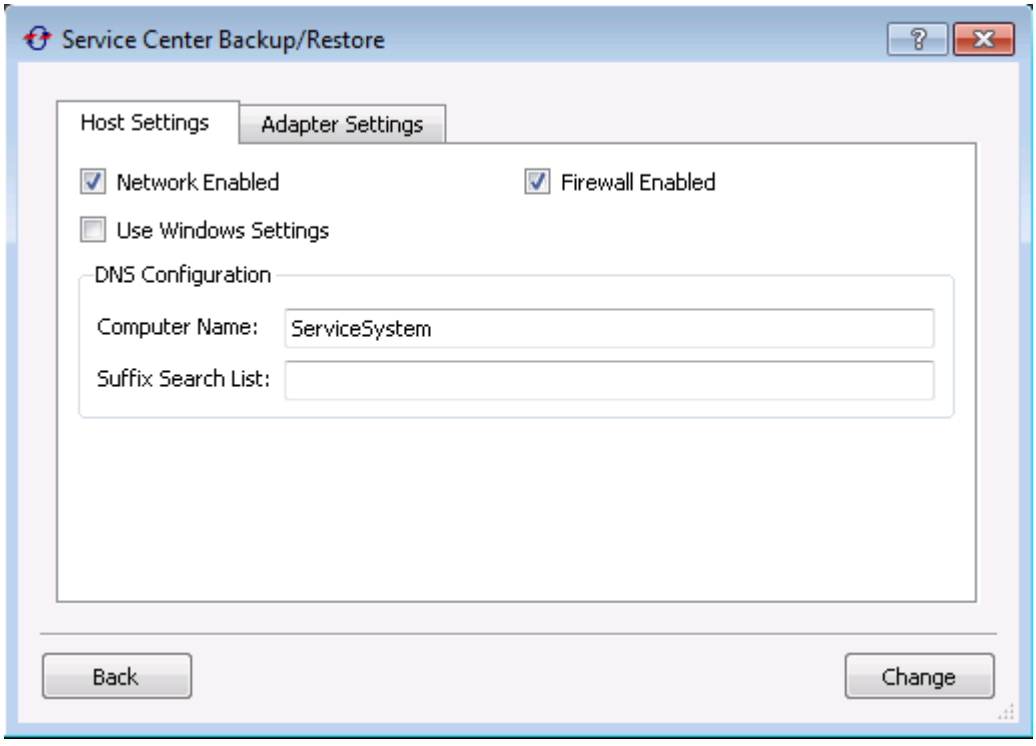



Figure 2-5 Host settings

Table 2-2 IP settings in the "Host Settings" dialog box

| Settings | Purpose |
|---|---|
| Network enabled | Activate or deactivate the network interfaces of the PCU 50.5. |
| Firewall enabled | Activate or deactivate the firewall of the PCU 50.5. |
| Use Windows Settings | Select whether the values set in the Control Panel are to be used. |
| DNS Configuration > Computer Name | Specify the DNS computer name. |
| DNS Configuration > Suffix Search List | Parameterize a search list for DNS suffix, e.g. "network.com".<br>The factory setting of the Ethernet interface "Local Area Connection" is ".local". |

# Install software and updates

# 3

## 3.1 Installing SINUMERIK products

### Overview

This chapter describes how to install additional software on the basis of the pre-installed PCE basic software or how to carry out an update.

The description below is based on the delivery condition of the hardware and software components.

The service desktop is, for example, used for the following tasks:

- Installing HMI system software
- Setting the running environment of the HMI system software
- Checking the hard disk or version
- Privilege for SIMATIC STEP 7

# 3.2 Installation via service desktop

## Installing additional software

The service desktop makes it possible to install system software or a software update. This mostly affects installation/update packages that are to be installed via the Windows network.

Installation can be performed in two ways:

- The installation/update package is stored in directory D:\INSTALL. When booting the PCU the next time, the installation/update process is automatically started while booting. Only when the installation or update process is completed does normal boot manager continue and, if necessary, HMI software started.
- The installation/update process can be started from the service desktop directly by executing the installation/update package.

## Using installation directories

Several installation directories can be set to enable an operator setup to be executed automatically. Installation directories include the subdirectory D:\Install and the directories listed in the [SetupDirs] section of the E:\Windows\System32\HMIServe.ini parameters file.

**The "D:\Install" installation directory is preset there.**

The key names contained within a section of the E:\Windows\System32\HMIServe.ini parameters file must be unique.

The installation directories are evaluated in the sequence described in the parameters file. If the parameters file is missing or it does not contain a [SetupDirs] section, the preset "D:\Install" installation directory is considered instead.

If the [SetupDirs] section contains installation directories, but not "D:\Install", the "D:\Install" directory is not considered within the context of the set installation directories.

When executing a setup using OpFile.txt, the issue of whether or not the setup requires a reboot once it has been completed is taken into account and displayed via a corresponding OpFile.txt entry. If a corresponding entry exists, a reboot is triggered. If there is a chain of setups to be executed one after the other, the reboot is performed once the final setup is complete.

## Installing with HMI Explorer

The "HMI Explorer" Windows program is available on the service desktop. When this program is called up, detailed version information relating to the HMI system software applications installed and to Windows XP is displayed.

Applications can be individually started or uninstalled via the HMI Explorer. Select the menu "Install" to display and install the available software packages.

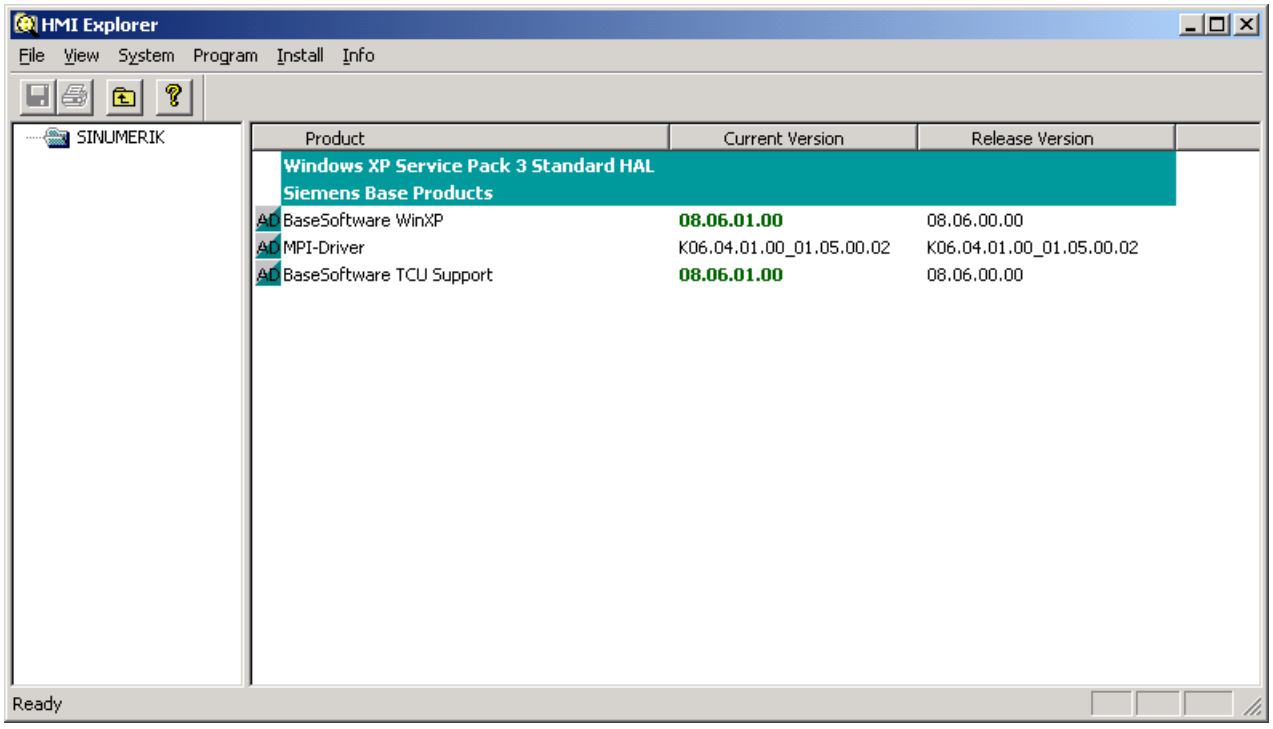



Figure 3-1 HMI Explorer (Example)

## Description of HMI Explorer

The properties dialogue box gives detailed information on installing the software product:

- **Information on the SINUMERIK product:**
  The "Info" dialogue box provides information on the selected SINUMERIK product:

| | |
|---|---|
| Current version: | Specifies which version of the SINUMERIK product is currently installed. The version is shown in long form. |
| Internal version: | Shows the current internal version number of this product. |
| Installation Date/Time: | Shows the installation date and time of the current version. |
| Installation path: | Displays the path for the main directory of the SINUMERIK product. |
| Start application: | Gives information on the path to the *.exe file, which launches the SINUMERIK product. |

- **Language of the product**
  The "Language" dialogue box lists the installed languages for the respective SINUMERIK product and provides information on the name of the installed language. If the language is not known, an abbreviation of its name is displayed. Known languages of HMI Explorer are German, English, Spanish, French and Italian. The version of the installed language is also displayed. Information is also given regarding the installation time and date.
- **History of the product**
  The "History" dialogue box shows information on the history of the SINUMERIK product. This dialogue box gives information on the release version, any service packs and hot fixes. The "release" entry is always available. The entries for "service pack" and "hot fix" only appear if they have been installed. Information on the "version", "internal version" and "installation date/time" is given for each entry on this list.
- **Component information**
  The "Components" dialogue box shows the information on the components accompanying a product:

| | |
|---|---|
| Component: | Component name |
| Version: | Internal version of the components |
| Path: | Path of the components |
| File: | *.exe file |
| Enable : | Shows whether or not the component is enabled |
| Description: | Description of components |
| Type: | Type of components |

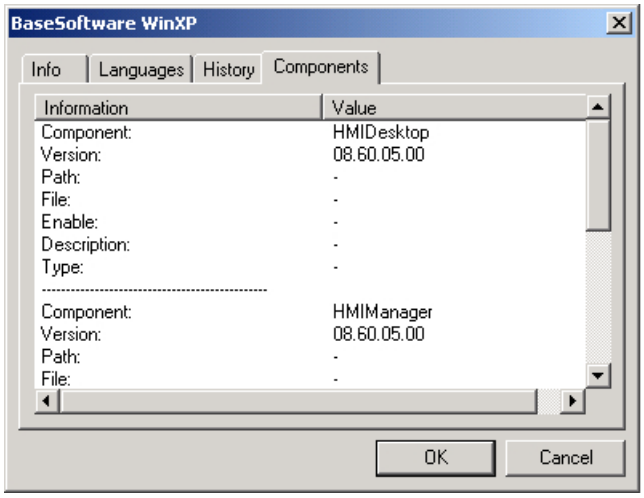



Figure 3-2 "Components" dialogue box (example)

# 3.3 How to install and authorize SIMATIC STEP 7

## Overview

SIMATIC STEP 7 V5.5 SP1 can also be installed on the PCU.

| | |
|---|---|
| Delivery item: | SIMATIC STEP 7 V5.5 SP1 |
| Components: | SIMATIC STEP 7 V5.5 SP1 and<br>AddOn for SINUMERIK 840D sl |
| Type of delivery: | 1 installation DVD |

**Note**

The "MPI driver" package available on the PCU is part of the HMI software and must not be uninstalled!

A network connection or a connection to a DVD drive is required.

**Recommendation:** Mouse port

## Installation under Windows XP

You must follow the installation sequence described here!

1. Boot up the PCU in the service desktop.
2. Access DVD (via network or external DVD drive) and call up SETUP.EXE in the root directory there.
3. The installation proceeds with operator prompting. The installation directory for STEP 7 should be changed to F:\... where the directory in F: can be freely selected.
4. The prompt for "Transfer license keys" must be answered with "No, transfer the license keys later." The licensing is done after the installation of SINUMERIK add-on. Once installation is complete, the PC will need to be rebooted. During boot up, select the service desktop again.
5. Switch to the Sinumerik_Add_on directory on the DVD and call SETUP.EXE. The installation proceeds with operator prompting. Once installation is complete, the PC will need to be rebooted. During boot up, select the service desktop again.
6. Start the link "STEP7 authorizing" on the service desktop. This authorizes STEP 7 and it can be started from the operating software (STEP 7 appears as its own operating area on the expansion bar of the area menu, protected with access level 3).
   The following entries are made automatically in F:\Add_on\oemframe.ini:

```
[s7tgtopx]
; with HMI Advanced: eliminate minimize/maximize buttons
; of the Step7 window
WindowStyle_Off=196608
; with HMI Advanced: switch to previous task when Step7 is
terminated
nSwitchToTaskAfterTermination= -2
```

   These entries may also need to be modified in OEM configurations.

# 3.4 How to install additional languages under Windows XP (DVD)

## Use

Use the SINUMERIK service pack recovery media WIN XP ProEmbSys SP3 to:

- Subsequently install Windows components
- Re-establish the delivery condition of the PCU **without application software**
- To install other languages for Windows XP

## Contents of the DVD

There are the following directories on the DVD:

| Directory | Contents |
|---|---|
| 1_WIN_Components | Windows XP ProEmbSys SP3<br>Windows XP ProEmbSys operating system, incl. SP3 for post-installation of software components that are no longer located on the PCU. |
| 2_XP_Base | Symantec Ghost image for PCU 50.3 and EBOOT<br>• Ghost image of the delivery condition of the PCU basic software Windows XP for PCU 50.3 without application software.<br>• Ghost image for creating an "Emergency Boot System" (identical to the directory D:\EBOOT on the PCU) |
| 3_MUI_1 | Chinese (simplified)<br>Traditional Chinese<br>Japanese<br>Korean<br>Romanian<br>Slovakian |
| 4_MUI_2 | Danish<br>German<br>French<br>Dutch<br>Italian<br>Spanish<br>Swedish |

| Directory | Contents |
|---|---|
| 5_MUI_3 | Brazilian Portuguese<br>Finnish<br>Polish<br>Russian<br>Czech<br>Turkish<br>Hungarian |
| EULA TERMS<br>Certificate of authenticity | Contained<br>Not contained |

## Installing languages

To install additional languages, proceed as follows:

1. If no DVD drive is directly connected via a USB port, the DVD can be accessed via a network to a released DVD disk drive via "Explorer" → "Tools" → "Map Network Drive". The letter G should be selected as the drive letter.
2. Select the directory with the appropriate language, the program "MUISETUP.EXE" starts. After accepting the licensing conditions, you can start the installation procedure with "Continue". A list of all of the pre-installed languages and the languages available on the DVD is displayed.
3. Now the desired languages can be installed/uninstalled by inserting or deleting a check mark in front of the language.
4. Other settings include:
   - Choice of language version for the standard user/new user
   - The language for programs without Unicode support must be set to "English (US)".
   - The font set must also be set to "English (US)".
5. After confirming with "OK," the installation begins. Many languages (e.g. Chinese) require system files that are also located on the DVD.
   If the installation drive is other than the recommended "G:" the "Windows XP Professional Service Pack 2 CD" or the "Windows XP Professional CD" may be required. The path name must then be changed accordingly.
6. After successful installation, a reboot may be requested depending on the language.

**Note**

- The choice can only be made from among the languages that were previously installed on the PCU.
- The new language of Windows XP only goes into effect if the user logs in again after the changeover or the PCU is turned off and then on again.
- The language setting of the operating software is independent of this. It is set independently of this under "Start-up" → "HMI" → "Change Language".

# Backing up and restoring data

# 4

## 4.1 Overview

You can perform installation and commissioning work on the PCU in theService Center, which is based on Microsoft Windows PE.

When the Desktop is active, configuration of the Service Center is possible via the "ServiceCenter Backup-Restore" (Page 65) icon.

Most of the functions of the Service Center are available in the main menu (Page 79).

In particular, you can create or restore disk images in the Service Center. The following options are available:

- Create a disk image of the SSD (Page 84)
- Restore a disk image of the SSD (Page 86)
- Create a disk image of a partition (Page 88)
- Restore a disk image of a partition (Page 90)

The Service Center uses the Symantec Ghost (Page 95) software, which you can also call separately.

## 4.2 Starting the Service Center

To start the Service Center, proceed in one of the following ways:

- To call up the Service Center on the SSD while the PCU is starting, select the "Booting Service System" during startup in the Windows boot menu.
- To call up the emergency boot system on a USB flash drive while the PCU is starting, set the service switch to E or press F12 during startup to show the boot selection.
- To switch to the Service Center while the Service Desktop (Windows 7) is active, click the "ServiceCenter Backup-Restore" icon on the Desktop and then click "Start ServiceSystem".

# 4.3 Functions of the Service Center

## Overview

In the upper section of the Service Center main menu, you can select one of five tasks. Buttons for additional functions are available in the lower section.

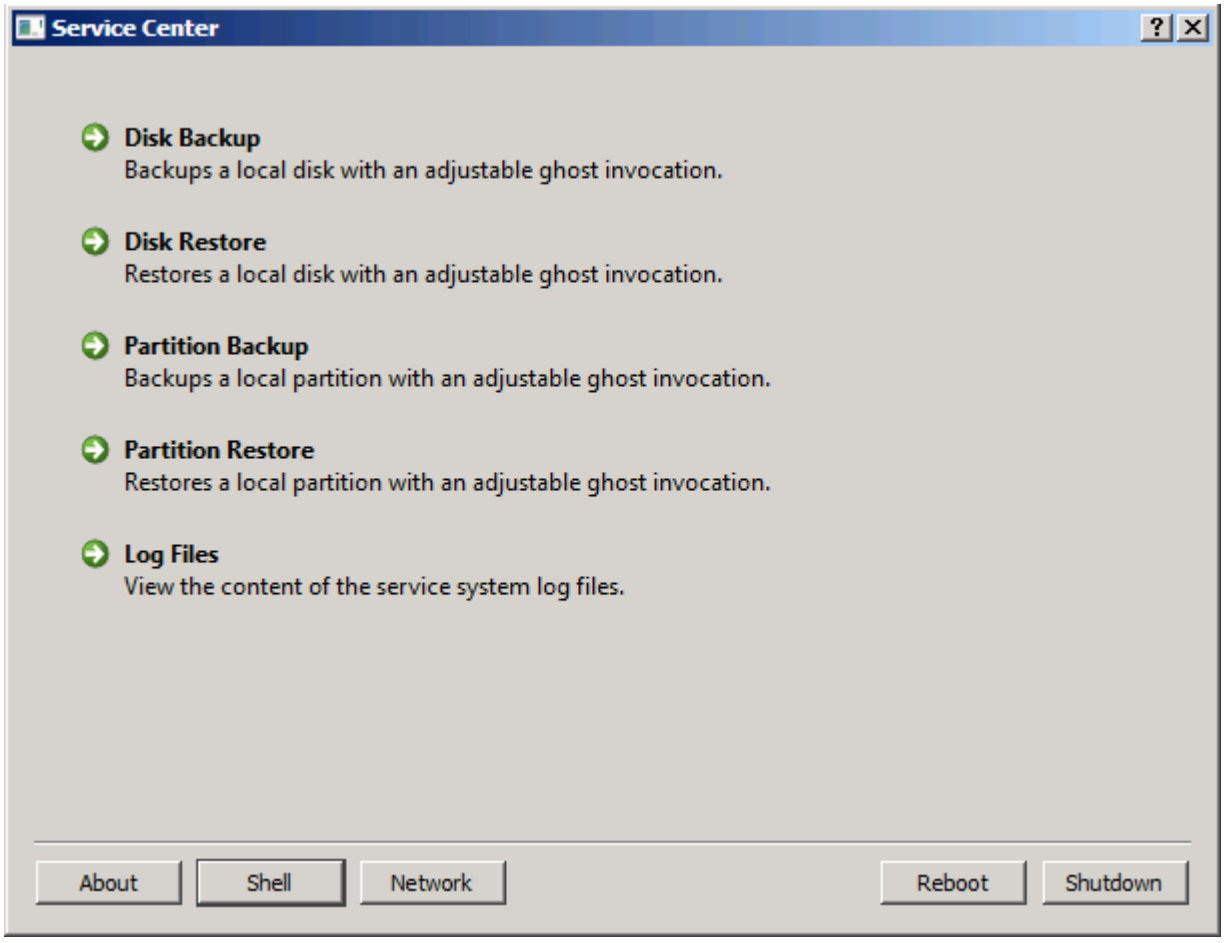



Figure 4-1 Service Center - Main menu

Table 4-1 Functions in the main menu of the Service Center

| Element of the window | Purpose |
|---|---|
| Disk Backup | Create a disk image of the local SSD.<br>With this command, you can create a disk image of the entire SSD (with all partitions). |
| Disk Restore | Restore the local SSD with an existing disk image.<br>With this command, you can restore the complete SSD (with all partitions) with a previously created disk image. |
| Partition Backup | Create a disk image of a partition on the local SSD.<br>With this command, you can create a disk image of an individual partition of the SSD. |
| Partition Restore | Restore a partition on the local SSD with a disk image.<br>With this command, you can restore an individual partition of the SSD with a previously created disk image. |

| Element of the window | Purpose |
|---|---|
| Log Files | View and save log files. |
| About | Viewing information about the Service Center. |
| Shell | Open the prompt.<br>With this command, you can enter commands for executing tasks on the PCU without using a graphical user interface. |
| Network | Making network settings (Page 81).<br>Any network settings you make in this dialog box are temporary. To change network settings permanently, switch to the Service Desktop. |
| Reboot | Restart the PCU. |
| Shutdown | Shut down the PCU. |

**Note**

**Data backup and restore with CompactFlash cards**

If there is a CompactFlash card in the slot of the PCU, you will not be able to back up or restore data via the Service Center.

Instead, start the "ghost32.exe" program via the prompt of the Service Center.

# 4.4 Network settings in the Service Center

You can make network settings in the main menu of the Service Center by clicking "Network":

- IP settings at the enterprise network interface X1 (Local Area Connection 2)
- IP settings at the plant network interface X2 (Local Area Connection)
- Configuration of the host
- Configuration of IP routing

**Note**

**Temporarily storing network settings in the Service Center**

Any network settings you make in this dialog box are immediately effective, but temporary.

To change network settings permanently, switch to the Service Desktop.

For further information, see Section Configuration of the Service Center (Page 65).

## Overview

### IP settings

You can make the following IP settings in the "Network Settings" dialog box:

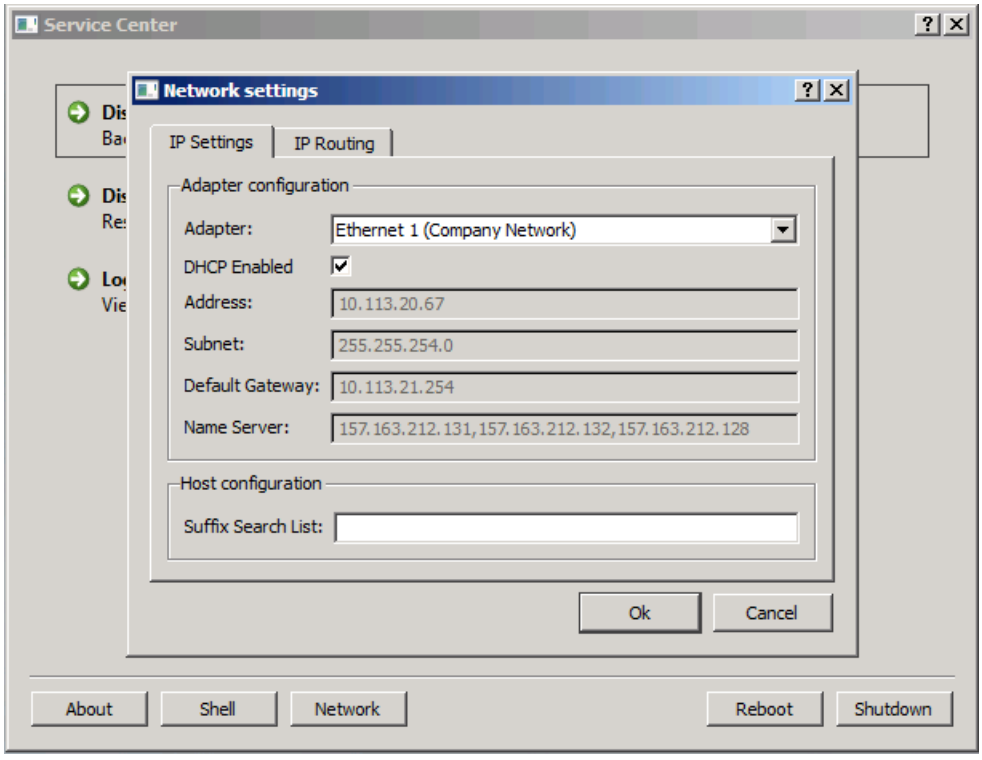



Figure 4-2 Service Center - IP settings

Table 4-2 IP settings in the "Network Settings" dialog box

| Section | Settings | Purpose |
|---|---|---|
| Adapter configuration | Adapter | Select the Ethernet interface that you want to configure.<br>All settings in Section "Adapter configuration" refer to the selected Ethernet interface. |
| | DHCP Enabled | Select whether the addresses are to be fetched dynamically from the DHCP server.<br>If the checkbox is deactivated, you must make the settings manually. |
| | Address | Set the IP address of the NCU.<br>You can specify an IP address from the following range:<br>192.168.214.250 - 254 |
| | Subnet | Specify a subnet mask, e.g. 255.255.255.0 |
| | Default Gateway | Set the IP address of the standard gateway.<br>A standard gateway creates a standard route in the IP routing table for all destinations that are not in the subnet. |
| | Name Server | Set the IP address of the name server.<br>A name server answers questions asked about a domain name zone using a DNS database. |
| Host configuration | Suffix Search List | Parameterize a search list for DNS suffix, e.g. "network.com".<br>The factory setting of the Ethernet interface "Local Area Connection" is ".local". |

### Routing settings

You can make the following routing settings in the "Network Settings" dialog box:

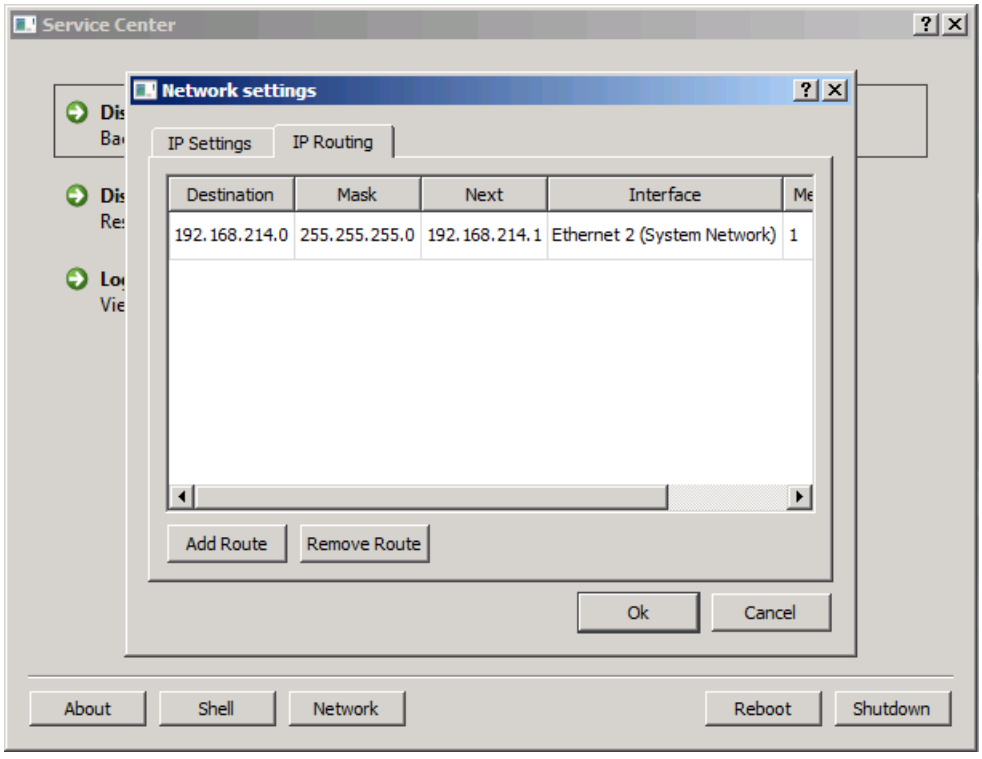



Figure 4-3 Service Center - IP routing

Table 4-3 Routing settings in the "Network Settings" dialog box

| Settings | Purpose |
|---|---|
| Add Route | Add a new route. |
| Remove Route | Delete a selected route. |
| Destination | Specify the network destination of the route as an IP address. |
| Mask | Subnet mask of the specified IP addresses. |
| Next | Next hop or forwarding IP address via which the network destination can be accessed. |
| Interface | Select an interface to which the configured IP routing applies:<br>• Enterprise network interface X1 (Local Area Connection 2)<br>• Plant network interface X2 (Local Area Connection) |
| Metric | Integer cost metric (1 to 9999). Required if several routes for sending a packet to a network destination are possible in the routing table. |

## See also

Release directory of the PC/PC in the network (Page 103)

Creating a shortcut to the network drive (Page 107)

Configuring routing in the network (Page 99)

# 4.5 Create a disk image of the SSD

You can create a disk image of the entire SSD in the Service Center with the "Disk Backup" function.

## Procedure

To save the SSD as a disk image, proceed as follows:

1. Start the Service Center, e.g. via the Windows boot menu.
   See also: Starting the Service Center (Page 78)
2. In the main menu of the Service Center, click "Disk Backup".
   The "Service Center - Disk Backup" dialog box opens.

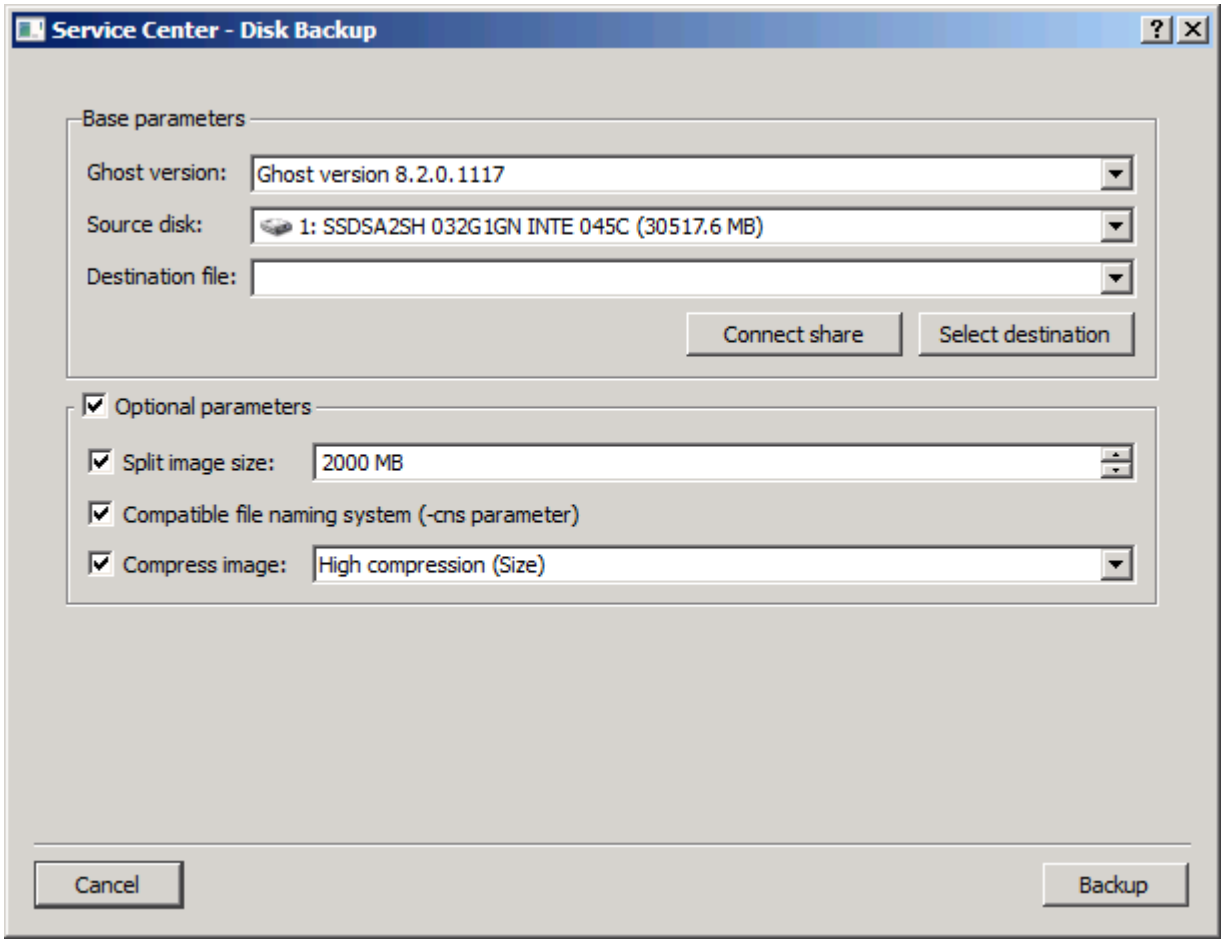



Figure 4-4 Service Center - Disk Backup

3. Make the required settings here:

| Section | Settings | Purpose |
|---|---|---|
| Base parameters | Ghost version | Select the Symantec Ghost version that is to be used when creating the disk image:<br>• Version for Windows 7: "Ghost version 11.5.x"<br>• Version for Windows XP: "Ghost version 8.2.x" |
| | Source disk | Select a data medium of which you want to create a data image. |
| | Destination file | Enter a destination directory and file name.<br>Disk images can be stored either locally or in the network. |
| Base parameters > Destination file | Connect share | Open the "Connect a Share" dialog box (Page 93).<br>You can set up a shared resource (e.g. a directory in the network), to which you can store the disk image. |
| | Select destination | Open the "Select Destination File" dialog box.<br>You can navigate through a local directory on the PCU 50.5 and enter a file name. |
| Optional parameters | Split image size | Split the disk image and select a split image size.<br>You can split the disk image into several files of a defined size.<br>Split the disk image size in the following cases:<br>• If the file system (e.g. FAT32) can only manage files of a certain size.<br>• If the disk image is to be archived onto multiple data media, for example, when using CDs. |
| | Compatible file naming system | Select the compatibility of the name scheme.<br>This option is required so that a disk image with older versions of the "PCU base software Windows XP" can be read in by Windows XP.<br>It is not possible to use a different operating system on a PCU 50.x than the preinstalled system (e.g. Windows XP on a PCU 50.5 Windows 7). |
| | Compress image | Activate Compression or select a compression level.<br>Higher compression results in a smaller file size, but increases the time needed to compress or decompress the data. |

4. Confirm your settings with "Backup" to start generation of the disk image.

# 4.6 Restore a disk image of the SSD

If your system is no longer functioning stably, you can restore it with a disk image.

**Note**

**Restoring replaces all current files**

When you restore the PCU from a disk image, all programs, settings, and files are completely replaced by the disk image. You cannot restore individual files or exclude individual files from restoration.

## Requirement

A disk image exists
See: Create a disk image of the SSD (Page 84)

## Procedure

To restore a disk image, proceed as follows:

1. Start the Service Center, e.g. via the Windows boot menu.
   See also: Starting the Service Center (Page 78)
2. In the main menu of the Service Center, click "Disk Restore".
   The "Disk Restore" dialog box opens.

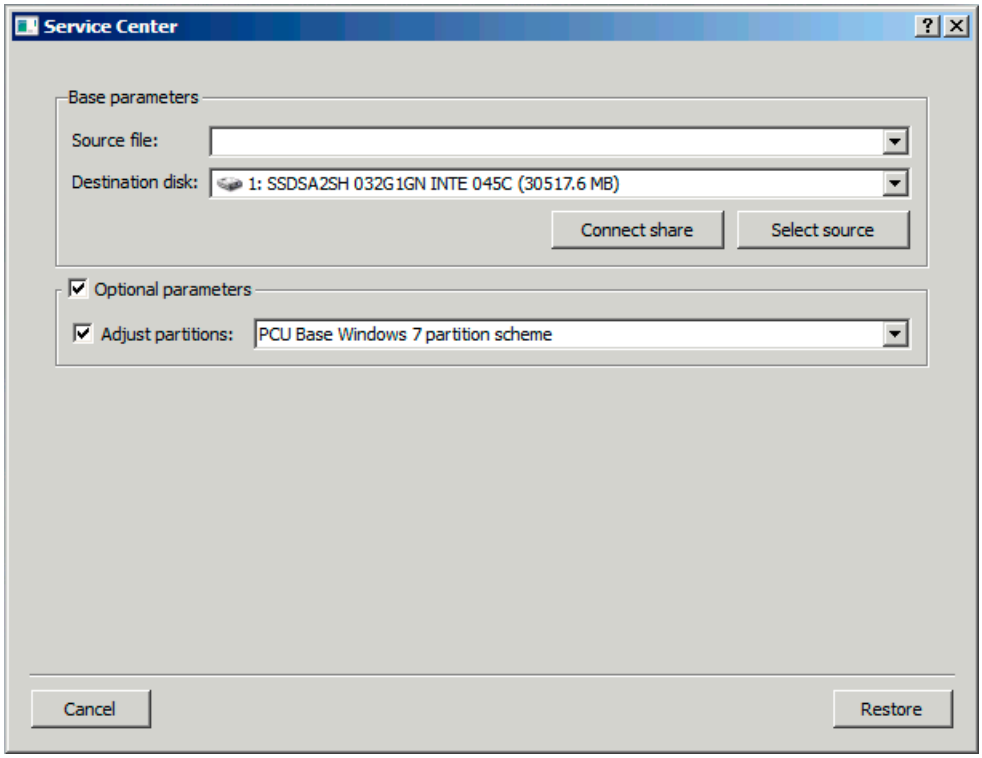



Figure 4-5 Service Center - Disk Restore

3. Make the required settings here:

| Section | Settings | Purpose |
|---|---|---|
| Base parameters | Source file | Select the disk image from which the SSD is to be restored.<br>You can use local disk images or disk images stored in the network. |
| | Destination disk | Select the SSD to be restored from the disk image. |
| Base parameters > Destination disk | Connect share | Open the "Connect a Share" dialog box (Page 93).<br>You can manage a shared resource (e.g. a directory in the network), to use a disk image stored on the network to restore the SSD. |
| | Select source | Open the "Select Source File" dialog box.<br>You can navigate through a local directory on the PCU 50.5 and select a disk image for restoration. |
| Optional parameters | Adjust partitions | Select partition scheme to be used on restoration of the data medium image:<br>• If the PCU base software Windows 7 is on the disk image, choose "PCU Base Windows 7 partition scheme".<br>• If the PCU base software Windows XP is on the disk image, choose "PCU Base Windows XP partition scheme".<br>It is not possible to use a different operating system on a PCU 50.x than the preinstalled system (e.g. Windows XP on a PCU 50.5 Windows 7). |

4. To start the restore process and entirely replace the existing data on the SSD with the disk image, confirm the settings with "Restore".

## Result

The restore process is started.

**Note**

**Troubleshooting if reading in of the disk image fails**

If data transfer is interrupted during the restore process, the PCU will have no executable system.

In this case, use a bootable USB flash drive with Service Center to read the disk image in again.

## 4.7 Create a disk image of a partition

You can create a disk image of an individual partition of the SSD in the Service Center with the "Partition Backup" function.

### Procedure

To save a partition of the SSD as a disk image, proceed as follows:

1. Start the Service Center, e.g. via the Windows boot menu.
   See also: Starting the Service Center (Page 78)
2. In the main menu of the Service Center, click "Partition Backup".
   The "Service Center - Partition Backup" dialog box opens.

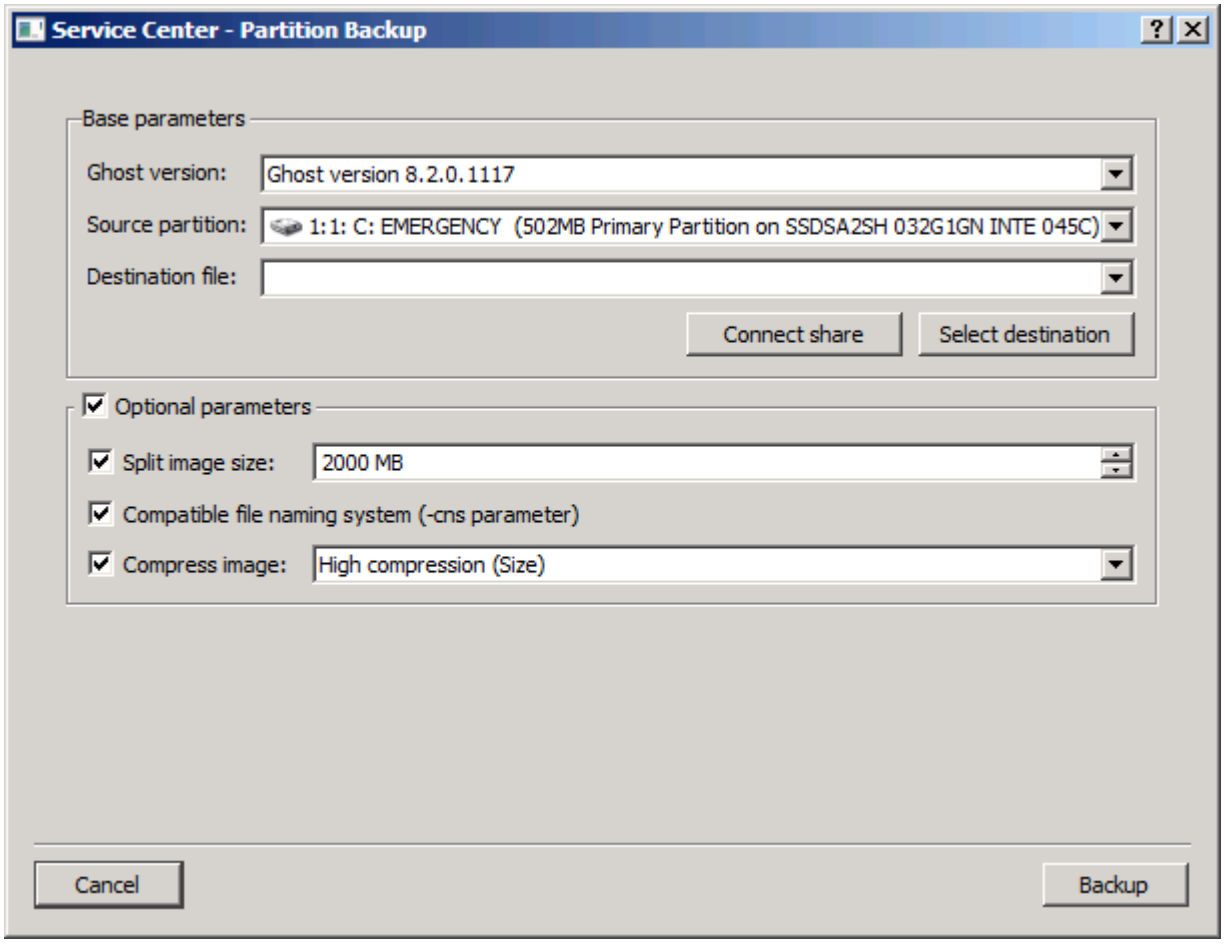



Figure 4-6 Service Center - Partition Backup

3. Make the required settings here:

| Section | Settings | Purpose |
|---|---|---|
| Base parameters | Ghost version | Select the Symantec Ghost version that is to be used when creating the disk image:<br>• Version for Windows 7: "Ghost version 11.5.x"<br>• Version for Windows XP: "Ghost version 8.2.x" |
| | Source partition | Select a data medium of which you want to create a data image. |
| | Destination file | Enter a destination directory and file name.<br>Disk images can be stored either locally or in the network. |
| Base parameters > Destination file | Connect share | Open the "Connect a Share" dialog box (Page 93).<br>You can set up a shared resource (e.g. a directory in the network), to which you can store the disk image. |
| | Select destination | Call the "Select destination file" dialog.<br>You can navigate through a local directory on the PCU 50.5 and enter a file name. |
| Optional parameters | Split image size | Split the disk image and select a split image size.<br>You can split the disk image into several files of a defined size.<br>Split the disk image size in the following cases:<br>• If the file system (e.g. FAT32) can only manage files of a certain size.<br>• If the disk image is to be archived onto multiple data media, for example, when using CDs. |
| | Compatible file naming system | Select the compatibility of the name scheme.<br>This option is required so that a disk image with older versions of the "PCU base software Windows XP" can be read in by Windows XP.<br>It is not possible to use a different operating system on a PCU 50.x than the preinstalled system (e.g. Windows XP on a PCU 50.5 Windows 7). |
| | Compress image | Activate Compression or select a compression level.<br>Higher compression results in a smaller file size, but increases the time needed to compress or decompress the data. |

4. Confirm your settings with "Backup" to start generation of the disk image.

## 4.8 Restore a disk image of a partition

If your system is no longer functioning stably, you can restore it with a disk image.

**Note**

**Restoring replaces all current files**

When you restore a partition from a disk image, all programs, settings, and files are completely replaced on this partition by the disk image. You cannot restore individual files or exclude individual files from restoration.

### Requirement

A disk image of a partition exists
See: Starting the Service Center (Page 78)

## Procedure

To restore a partition from a disk image, proceed as follows:

1. Start the Service Center, e.g. via the Windows boot menu.
   See also: Create a disk image of a partition (Page 88)
2. In the main menu of the Service Center, click "Partition Restore".
   The "Partition Restore" dialog box opens.

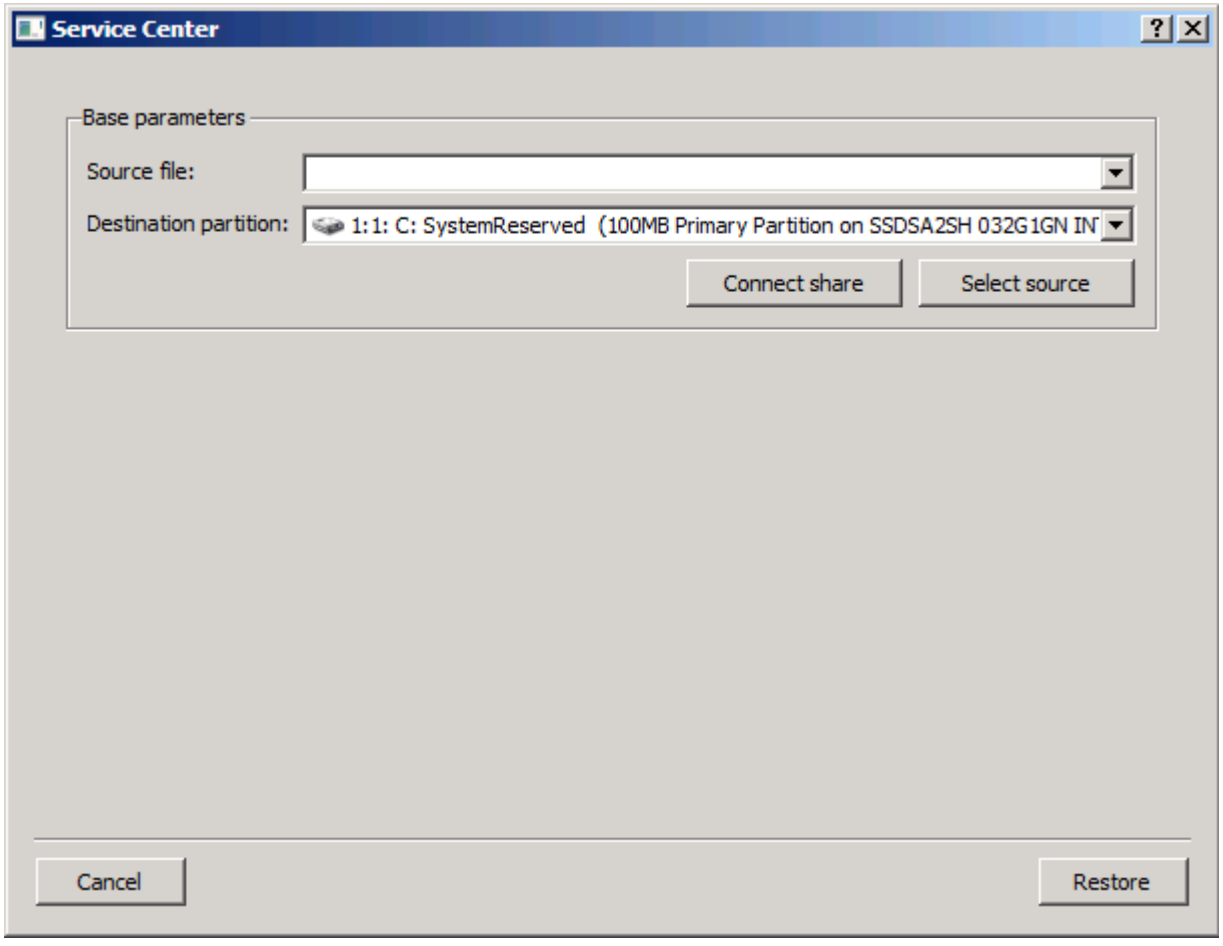



Figure 4-7 Service Center - Disk Restore

3. Make the required settings here:

| Section | Settings | Purpose |
|---|---|---|
| Base parameters | Source file | Select the disk image from which the partition is to be restored.<br>You can use local disk images or disk images stored in the network. |
| | Destination partition | Select the partition to be restored from the disk image. |
| Base parameters > Destination partition | Connect share | Open the "Connect a Share" dialog box (Page 93).<br>You can manage a shared resource (e.g. a directory in the network), to use a disk image stored on the network to restore the SSD. |
| | Select source | Call the "Select source file" dialog.<br>You can navigate through a local directory on the PCU 50.5 and select a disk image for restoration. |

4. To start the restore process and entirely replace the existing data on the partition with the disk image, confirm the settings with "Restore".

## Result

The restore process is started.

**Note**

**Troubleshooting if reading in of the disk image fails**

If data transfer is interrupted during the restore process, the PCU will have no executable system.

In this case, use a bootable USB flash drive with Service Center to read the disk image in again.

# 4.9 Network settings in the Service Center

To manage a disk image in the network, you can include a shared network directory on the PCU 50.5 as a drive.

In the Service Center, you configure network drives in the "Connect a share" dialog box. You can call this dialog box directly from the relevant dialog boxes to both create (Page 84) and restore (Page 86) disk images.

## Requirement

The network directory is shared with the relevant user accounts in the domain and on the PCU.
See also: Release directory of the PC/PC in the network (Page 103)

## Overview

The setting options in the "Connect a share" dialog box have the following meaning:

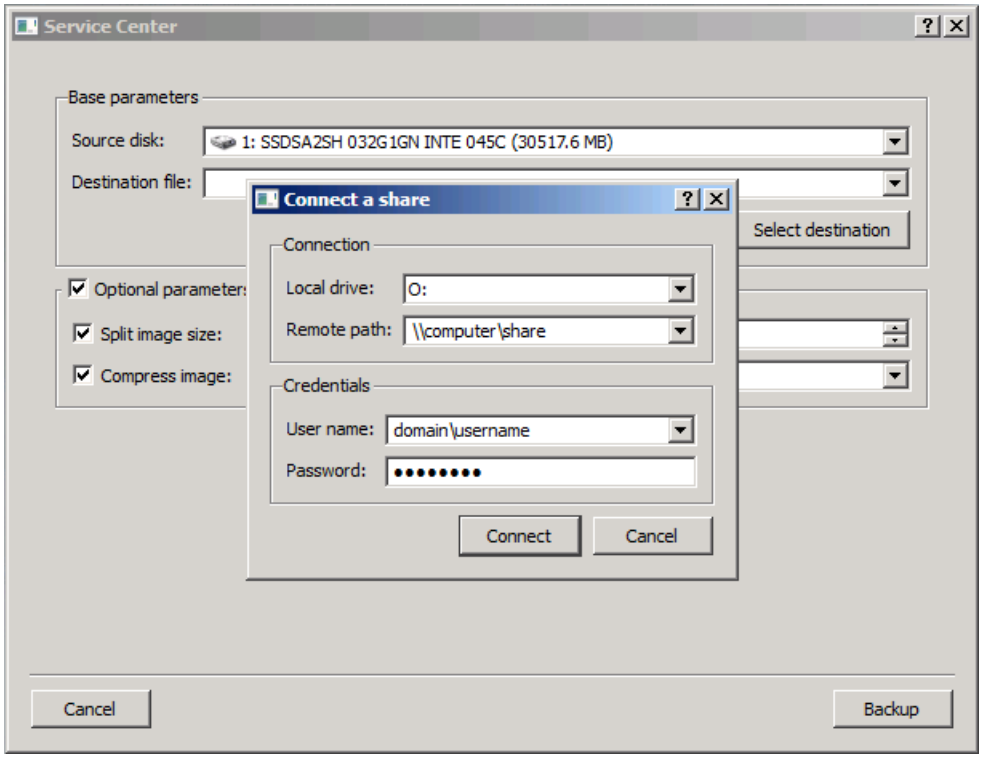



Figure 4-8 "Connect a Share" dialog box

Table 4-4 Settings in the "Connect a Share" dialog box in the Service Center

| Section | Settings | Purpose |
|---|---|---|
| Connection | Local drive | To select drive letters under which the network directories are to be accessible on the PCU 50.5. |
| | Remote path | Specify the device in the network and share name of the resource.<br>Syntax: `\\Name of the server\share name of the network directory`<br>Example: `\\Backup_Server\PCU_Backup` |

| Section | Settings | Purpose |
|---|---|---|
| Credentials | User name | Enter the user account under which the resource in the network can be accessed.<br>If you have shared the resource on the server for certain users only, you must enter one of these user accounts here. |
| | Password | Enter the password of the user account. |

## Further information

You will find information about releasing directories in the network and network drives in the following sections:

- Release directory of the PC/PC in the network (Page 103)
- Creating a shortcut to the network drive (Page 107)

## 4.10 Starting Symantec Ghost directly

With the "Symantec Ghost" software you can store the entire contents of an SSD as a disk image. Disk images can be kept on various storage media for subsequent restoration of the data on the SSD.

Symantec Ghost is supplied from the factory on replacement SSDs for the PCU and on the SSD of the entire PCU. Symantec Ghost is used by the components of the PCU Base Software to back up or restore data.

You can also start Symantec Ghost directly from the prompt to execute the data backup tasks from the user interface of Ghost. This is necessary, for example, when using CompactFlash cards.

### Procedure

To start Symantec Ghost directly, proceed as follows:

1. Call the prompt from the Service Center or the Service Desktop.
2. Enter "ghost.32.exe" and confirm the entry.

### Further information

More information is available in the Internet on website Ghost.com (http://www.ghost.com).

# 4.11 Bootable USB flash drive

## Use

You can generate a bootable Service Center (Emergency Boot System) based on Microsoft Windows PE on a USB flash drive as preventive measure against a defect

The Ghost image is located on the SSD under: `D:\Eboot\eboot.gho`

Using the same procedure, you can create a Service Center on a PG/PC. For this, `Symantec Ghost` must be installed on the PG/PC.

## Precondition

- The Windows 7 Service Desktop is active
- You have a USB flash drive (e.g. SIMATIC PC USB flash drive)

## Procedure

To create a bootable USD flash drive, proceed as follows:

1. Connect the USB flash drive to a USB interface of the PCU.
2. Start Symantec Ghost, for example, by entering `Ghost32.exe` in the search bar in the start menu.
3. In Symantec Ghost, select the "From Image" command from the "Local > Disk" menu.
4. Select `D:\Eboot\eboot.gho` as the source and the USB flash drive as the destination.

## Result

The Emergency Boot System is installed on the USB flash drive and the data medium is renamed to "EBOOT." The service system on the bootable USB flash drive can now be used.

# 4.12 Operating a service PC/PC in the network

## 4.12.1 Overview

### System network

In the following applications, you will need, for example, a connection between the PCU and a PG/PC in the system network:

- Save the disk image of the PCU-SSD to a PG/PC
- Restore the SSC of a PCU via the DVD drive of a PG/PC
- Start up the replacement SSD
- Install the software on the PCU via the DVD drive of a PG/PC

### Company network

If you want to connect the service PG/PC via a company network (Eth 1), contact your network service center.

## 4.12.2 Connection options in the system network

### Overview

The following figures show the typical connection options in the system network:

- PCU to "Eth 2" with service PG/PC, directly, using a crossover Ethernet cable
- PCU to "Eth 2" with service PG/PC, via a switch, using a 1:1 Ethernet cable

Table 4-5 Configuration images: Meaning of the connections

| Symbol | Meaning |
|---|---|
|  | Eth 1 as a DHCP client |
|  | Eth 2 with a fixed IP address |
|  | 1:1 Ethernet cable |
|  | Crossover Ethernet cable |

## Configuration with PG/PC directly to PCU

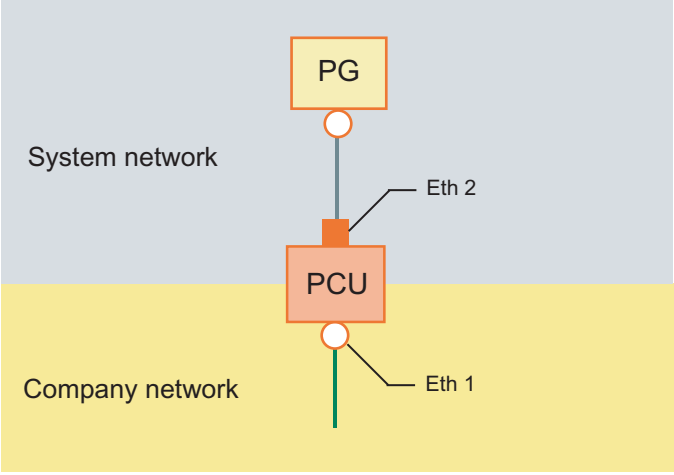



Figure 4-9 Connecting a PG directly to a PCU

## Configuration with PG/PC and switch to PCU

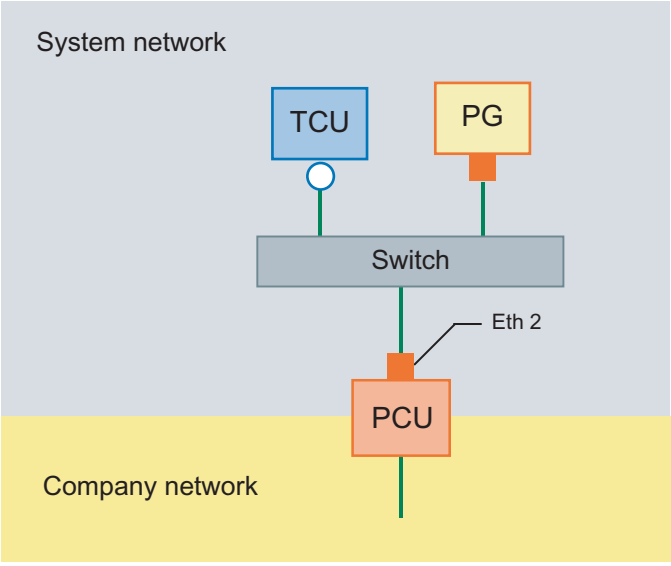



Figure 4-10 Connecting a PG via a switch to a PCU

**Note**

If a PCU is switched off and on again without its own OP/TP including TCU, and if the PCU boot-up is intended to be performed from the service system (EBS), an external VGA monitor and keyboard are needed to operate the PCU.

An external VGA monitor and keyboard are not needed if the TCUs are operating and the PCU is not switched off with the EBS during booting.

### Basic procedure

On the PG/PC with Windows 7:

- Connecting a PG/PC to a PCU according to one of the configurations shown above.
- The network protocol used is: TCP/IP.
  TCP/IP is already preconfigured on the PCU.
- Setting up IP addresses on the same subnetwork.
- Releasing a directory on the PG/PC for network access.

On the PCU in the Service Center (WinPE):

- On the PCU start the Service Center (WinPE).
- Establish a network connection to the shared directory of the PG/PC.
  See: Functions of the Service Center (Page 79)

## 4.12.3 Configuring routing in the network

### Overview

If you have connected a PG/PC at interface X127 of the NCU, configure the network routing to establish the connection between the PG/PC and the PCU.

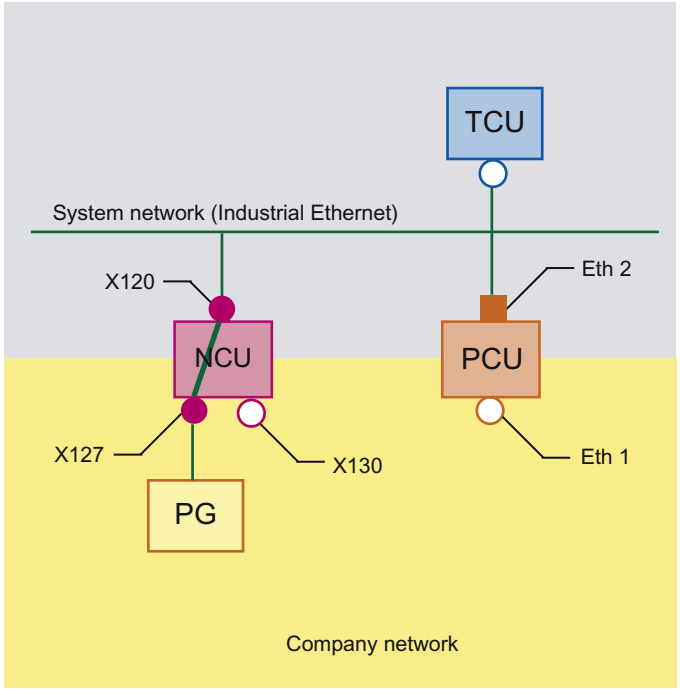



Figure 4-11 A networking example using IP routing

## Conditions

The following conditions must be met in order to use this function:

- The PG must be connected to interface X127 of an NCU in the system network.
- The NCU and PCU must be connected via the system network.
- On the NCU, the routing is activated in one of the following ways:
  - In SINUMERIK Operate 4.5 SP2 or higher, the routing is configured in image "Settings of the system network".
  - In the configuration file basesys.ini, key EnableSysNetToIBNForwarding=1 is set.
- If you wish to access a network directory, it must be shared.
- The user account in question is known to the PG/PC.
- The Service Center is active.

## Procedure

To configure the IP routing in the Service Center, proceed as follows:

1. In the main menu of the Service Center, click "Network".
2. In the "Network Settings" dialog box, click the "IP Routing" tab.

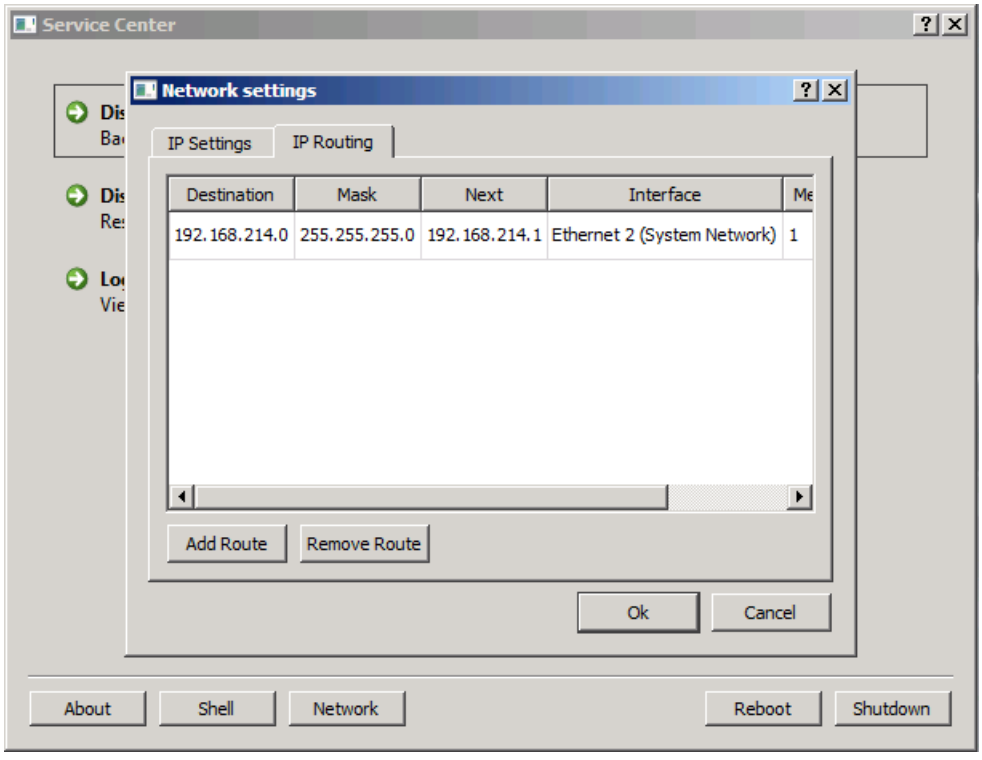



Figure 4-12 Service Center - IP routing

3. Configure the necessary routes. See also: Network settings in the Service Center (Page 81)

### Result

The routing has been configured and the PG/PC and PCU are connected via the NCU.

Now, perform the desired steps, for example, a Restore a disk image of the SSD (Page 86).

## 4.12.4 Configure the network settings of the PG/PC

Once you have connected a PG/PC in the network, make the network settings on the PG/PC.

The basic procedure for setup under Windows 7 or Windows XP is described below. Detailed information from Microsoft is provided in Windows under "Help and Support" or in the Internet: Microsoft Windows support (http://windows.microsoft.com/en-US/windows/windows-help#windows=windows-7)

### Precondition

- The PG/PC is networked with the PCU in one of the following ways:
  - The PG/PC is connected to the PCU with an Ethernet cable (either directly or via a switch)
    See: Connection options in the system network (Page 97)
  - The PG/PC is networked with the PCU via an interface on the NCU and the routing is configured.
    See: Configuring routing in the network (Page 99)
- A current Microsoft Windows operating system is installed on the PG/PC
- The PG/PC is switched on

## Procedure

To configure the network settings of the PG/PC for connecting a PCU in the system network, proceed as follows:

1. Open the "network connections" window:
   - To do this under Windows XP: in the "Start > Control Panel > Network and Internet Connections" menu, choose "Network Connections."
   - To do this under Windows 7: in the Control Panel in category "Network and Internet > View network status and tasks," choose "Change adapter settings."
2. Right-click on the network adapter you want to configure, then choose "Properties" from the shortcut menu.
   This opens the "Properties of <connection>" dialog box.

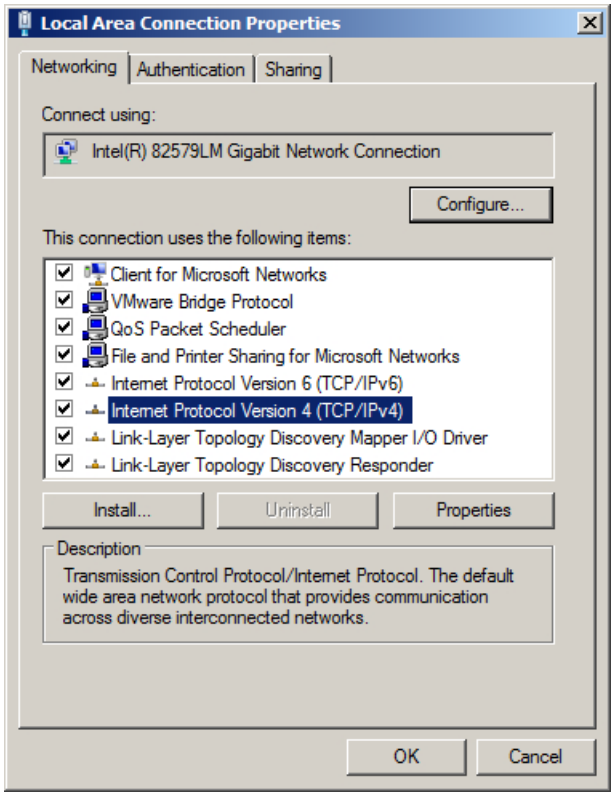



Figure 4-13 Properties of <connection>

3. Make the following settings:
   - Click "File and Printer Sharing for Microsoft Networks."

– To change the IP address of the PG/PC, double-click "Internet Protocol Version 4 (TCP/IPv4)", then configure the settings as follows:

| Section | Element | Settings |
|---|---|---|
| Use the following IP address | IP address | The following IP addresses in the system network can be used for the PG/PC:<br>• 192.168.214.250<br>• 192.168.214.251<br>• 192.168.214.252<br>• 192.168.214.253<br>• 192.168.214.254 |
| | Subnet mask | • 255.255.255.0 |

### Result

The network settings of the PG/PC are now configured for use with the PCU 50.5.

## 4.12.5 Release directory of the PC/PC in the network

If you want to, for example, store disk images on a PG/PC that you have linked to the PCU via the network, set up a network drive.

As soon as it has been set up, a new drive will appear on the PCU, which is really a directory on the PG/PC.

The basic procedure for setup under Windows 7 and Windows XP is described below. Detailed information from Microsoft is provided in Windows under "Help and Support" or in the Internet: Microsoft Windows support (http://windows.microsoft.com/en-US/windows/windows-help#windows=windows-7)

### Requirement

- The PG/PC is switched on.
- There is sufficient space on the PG/PC for saving, for example, disk images from the PCU.
- The user accounts to which you want to grant share rights are known in the domain or on the PCU.

### Procedure

Proceed as follows to share a directory for use in the network:

1. On the PG/PC, in the Control Panel call up Computer Management.
2. Under "System > Shared Folders" select the file "Shares."
3. In the "Action" menu, click the "New File Share..." command.
   The "Create A Shared Folder Wizard" opens.

4. Under step "Name, Description, and Settings," make a note of the Share path, which is a combination of the computer name and the share name. You will have to specify this name when you create a link to the network drive on the PCU.

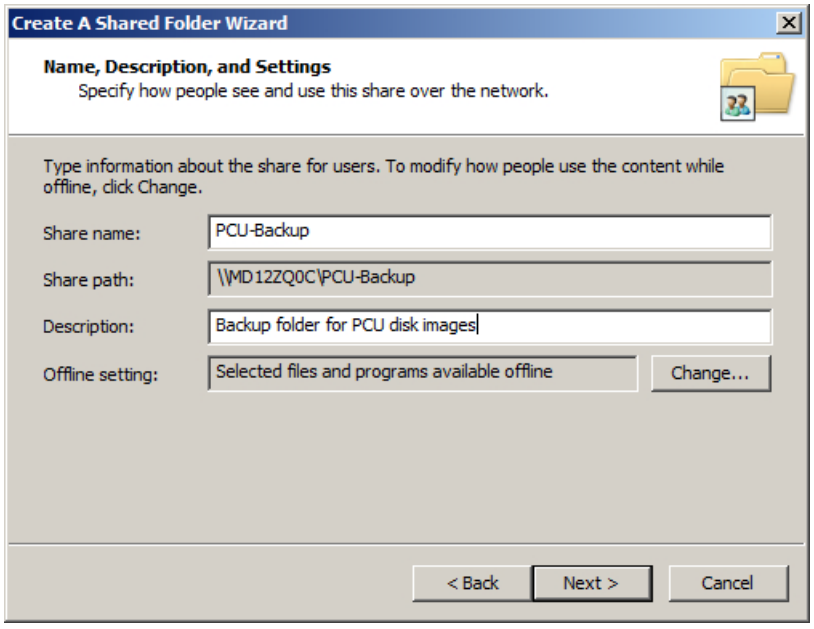



Figure 4-14 Create A Shared Folder Wizard - Name, Description, and Settings

5. Under step "Shared Folder Authorizations," select the "Customize authorizations" radio button, then click "Custom..."

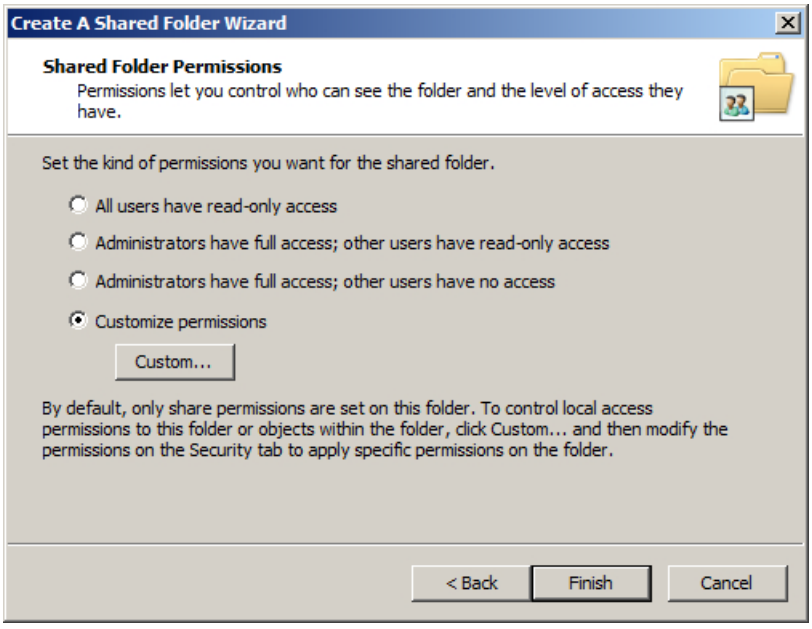



Figure 4-15 Create a Shared Folder Wizard - Shared Folder Authorizations

The "Customize Authorizations" dialog box opens.

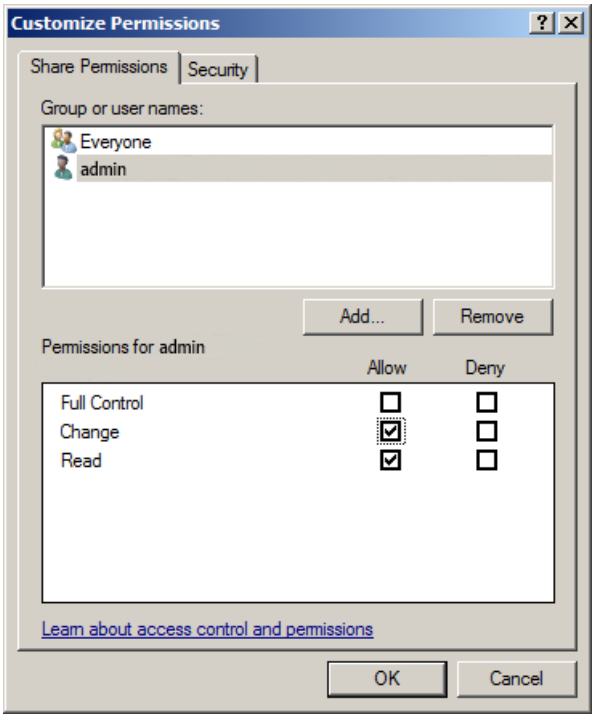



Figure 4-16 "Customize Authorizations" dialog box

6. In section "Group or user names," click "Add..."
   The "Select Users, Computers, Service Accounts, or Groups" opens.

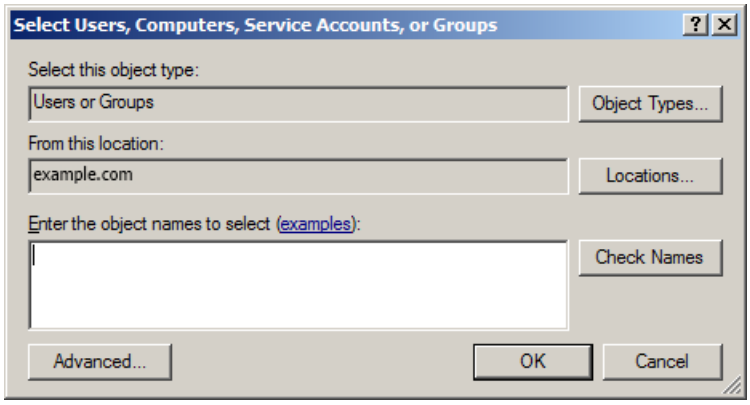



Figure 4-17 "Select Users, Computers, Service Accounts, or Groups" dialog box

7. In section "Authorizations for <user name>," under column "Allow," click the "Change" checkbox.
8. Confirm these settings in the "Customize PermiAuthorizations" dialog box with "OK."
9. In the next steps of the wizard, make all the settings you require and confirm the share with "Finish".

## 4.12.6 Release drives C: and D: of the PCU 50.5 in the network for administrator accounts

You can release local drives (C$, D$) of the PCU 50.5 in the network for administrator accounts

You can then access these PCU 50.5 drives, for example from a PG/PC in the network if you use an administrator user account that is available on the PCU 50.5.

### Procedure

1. Enter "regedit" in the search field of the registry.
   The registration editor opens.
2. Navigate to the following folder in the folder structure: HKEY_LOCAL_MACHINE \SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
3. Right-click in the working area and select the "New > DWORD (32-bit) Value" command in the shortcut menu.
4. In the "Name" field, enter the "LocalAccountTokenFilterPolicy" designation.
5. Double-click the key and enter "1" as "Value".
6. Confirm the input, close the registration editor and restart the PCU 50.5.

### Result

The registry entry has been added and the drives are available in the network after the restart.

However, you cannot change the access rights, i.e. access is only possible via an administrator account that is known on the PCU 50.5.

To undo the release, delete the created key in the registration editor.

## 4.12.7 Creating a shortcut to the network drive

To be able to access the shared directory on the PG/PC from the PCU, after you have connected and configured the PG/PC in the network, you must set up the folder shared on the PG/PC on the PCU as a network drive.

As soon as it has been set up, a new drive will appear on the PCU, which is really a directory on the PG/PC.

The basic procedure for setup under Windows 7 and Windows XP is described below. Detailed information from Microsoft is provided in Windows under "Help and Support" or in the Internet: Creating a shortcut to a network drive (association) (http://windows.microsoft.com/en-us/windows7/create-a-shortcut-to-map-a-network-drive)

### Precondition

- The PG/PC is switched on.
- The PCU is switched on and the Service Desktop is active.
- The folder on the PG/PC is shared for the relevant user accounts on the domain and on the PCU.

### Procedure

Proceed as follows to connect a shared directory as a network drive:

1. Start the Windows Explorer and click the "Map network drive" button.
2. Make the following settings in the "Map network drive" dialog box:

| Settings | Purpose |
|---|---|
| Drive | To select drive letters under which the network directories are to be accessible on the PCU 50.5. |
| Folder | Specify the device in the network and share name of the resource.<br>Syntax: `\\Name of the server\share name of the network directory`<br>Example: `\\Backup_Server\PCU_Backup` |
| Reconnect at logon | Select this check box to automatically connect the network drive on booting. |
| Connect using different credentials | Select this check box if you have not shared the directory in the network with the user for whom you would like to set up the network drive.<br>In the next dialog box "Enter Network Password," enter the log-on information of a user with whom the network directory is shared. |

3. Confirm your settings with "Finish."

## See also

Microsoft Windows support (http://windows.microsoft.com/en-US/windows/windows-help#windows=windows-7)

## 4.13 Starting up the replacement SSD

Information about replacing and installing an SSD on the PCU is provided in the Operator Components and Networking Manual.

**Note**

**Troubleshooting if reading in of the disk image fails**

If reading in of the data is interrupted, the PCU will have no executable system.

In this case, use a bootable USB flash drive with Service Center to read the disk image in again.

### See also

Restore a disk image of the SSD (Page 86)

# Service and diagnostics 5

## 5.1 PCU Hardware Diagnostics

### Intended use

The PCU hardware supports the diagnostics of important system components via an integrated "safecard", which is designated as a Safecard-On-Motherboard (SOM).

These diagnostic functions are only evaluated by systems with HMI Advanced. The fault statuses of the hardware are reported in the form of alarms via the operator interface of HMI Advanced. This allows for visualization of the data in HMI Advanced and external evaluation.

### Monitored Parameters

The following physical parameters of the PCU hardware are monitored:

- CPU temperature
- Housing temperature
- I/O chip temperature
- Speed of the two housing fans
- S.M.A.R.T - status of the hard disk

### Logging faults

The PCU hardware monitor logs all hardware faults in the Windows event log so that the faults can be output even without installing an HMI application.

The alarms are output in the log under "Control Panel" → "Administrative Tools" → "Event Viewer".

### See also

- AUTOHOTSPOT
- Operator Components and Networking Manual: Chapter, spare parts
- Description of the alarms: SINUMERIK 840D sl Diagnostics Manual

# 5.2 Evaluation of the 7-segment display

## Intended use

The 7-segment display is intended for PCU diagnostics in operation without a local OP (known as "Headless mode"). If an error is detected during booting, a local display is needed for subsequent error analysis of the connection.

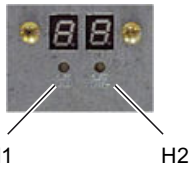



Figure 5-1 7-segment display with LEDs

The two displays have the following function:

- Segment display H1
  Segment display H1 and the associated LED are assigned to the PCU basic software. Status codes are output during system booting, normal operations and shutdown. Compatibility in relation to the NCU modules is taken into account.
- Segment display H2
  Segment display H2 and the associated LED are assigned to the application software.

## Meaning of status codes

- During system power-up:

| LED H1 | LED H2 | 7-segment display | | Status | |
|---|---|---|---|---|---|
| Orange | Orange | Output of BIOS post codes | | After switching on the system | |
| Orange | Off | | | After cycling the BIOS | |
| Orange | Off | | | After starting Windows | |
| | | 1 | 0 | Start Windows | Load device drivers needed for the Windows start |
| | | 2 | 0 | PCU hardware service | The PCU hardware service has been started. |
| | | 5 | 0 | Network | Wait for network interfaces to be ready. |
| | | 8 | 0 | TCU support test step 1 | Wait for FTP server to start |
| | | 9 | 0 | TCU support test step 2 | Wait for boot server to start for TCU network boot and for TCU hardware service to start |
| | | A | 0 | TCU support test step 3 | Wait for VNC server to start |

| LED H1 | LED H2 | 7-segment display | | Status | |
|---|---|---|---|---|---|
| | | B | 0 | HMI manager | Wait for HMI manager to start |
| | | E | 0 | PCU base software | General fault: see Windows "Event Viewer" |

- During operation:

| LED H1 | LED H2 | 7-segment display | | Status | |
|---|---|---|---|---|---|
| Flashing green | -- | 0 | 0 | OK | |
| Flashing red | -- | | | Error: | |
| | | 1 | 0 | Temperature alarm | Housing or CPU temperature; temperature above limit value (SOM error) |
| | | 2 | 0 | Fan alarm | Housing or CPU: Low fan speed or fan failure (SOM error) |
| | | 3 | 0 | Hard disk alarm | S.M.A.R.T hard disk error |
| | | 6 | 0 | | Failure of VNC server or VNC server service stopped |

- During shutdown

| LED H1 | LED H2 | 7-segment display | | Status |
|---|---|---|---|---|
| Flashing red/ green | -- | 5 | 0 | Shutdown active |

# 5.3 Setting of the service switch (PCU 50.5 only)

## Intended use

The "PCU Hardware Service" component as part of the PCU Base software provides a function to read-out the switch positions.




Figure 5-2 Rotary coding switch (hexadecimal)

## Meaning of switch positions

Only the settings of the service switch that are relevant for the commissioning engineer are listed:

| Switch setting | Mode of operation | Function |
|---|---|---|
| **0** | --- | Normal operation (default setting) |
| **6** | Users | Reserved for additional software |
| **9** | Service | Headless operation without OP/TP |
| **A** | Service | Firmware recovery |
| **C** | Service | Checking |
| **D** | Service | Diagnostics |
| **E** | Service | Booting from the emergency boot system |

# 5.4 Enabling/disabling error log during boot up

## Application

Each time the system is booted up, information is written to a block in the file D:\$$Base.log, which contains the date, time and nature of administrative interventions.

Information about the booting controlled by the HMI Manager, can be displayed on the screen and written to log file D:\$$Base.log.

## Output in log file

The output is controlled by registry values:

- **Type of information**
  The type of information to be output is set via registry value “HKLM\SOFTWARE\Siemens\SINUMERIK\Basic software\<version>\HMI Manager\InfoLevel”:

**InfoLevel (DWORD)**

= 1: Mandatory information is output (default setting)

= 2: Mandatory and supplementary information is output

= 3: Mandatory, supplementary and trace information is output
(InfoLevel <= 0 is treated as InfoLevel == 1;
InfoLevel <= 3 is treated as InfoLevel == 3)

- **Outputting information to the screen**
  Whether the information is also to be displayed on the screen (as well as being output to the log file) is set via registry value 'HKLM\SOFTWARE\Siemens\SINUMERIK\Basic software\<version>\HMI Manager\ShowInfo':

**ShowInfo (DWORD)**

= 0: Displays the mandatory information,
no display of the supplemental and trace information

= 1: Mandatory, supplementary and trace information is displayed (default)

# 5.5 How to search for stations within the system network

## Intended use

The VNC scanner is used within the system network. The scanner is primarily used to identify devices containing an active VNC server, e.g. NCU or PCU.

The VNC scanner offers the option of connecting directly to the VNC server and, for example, operating an HMI application. The integrated VNC viewer is used to do this.

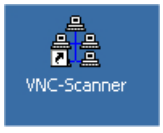
Furthermore, it enables you to call up a list of all the network devices, thereby creating an overview of the system network.

**Note**

The VNC scanner is available on the Service Desktop of the PCU and on the PCU base software DVD for installation on a PG/PC.

You require a mouse to use this program.

## Operating "VNC scanner"




1. You can launch the "VNC Scanner" program via this link on the Service Desktop of the PCU .

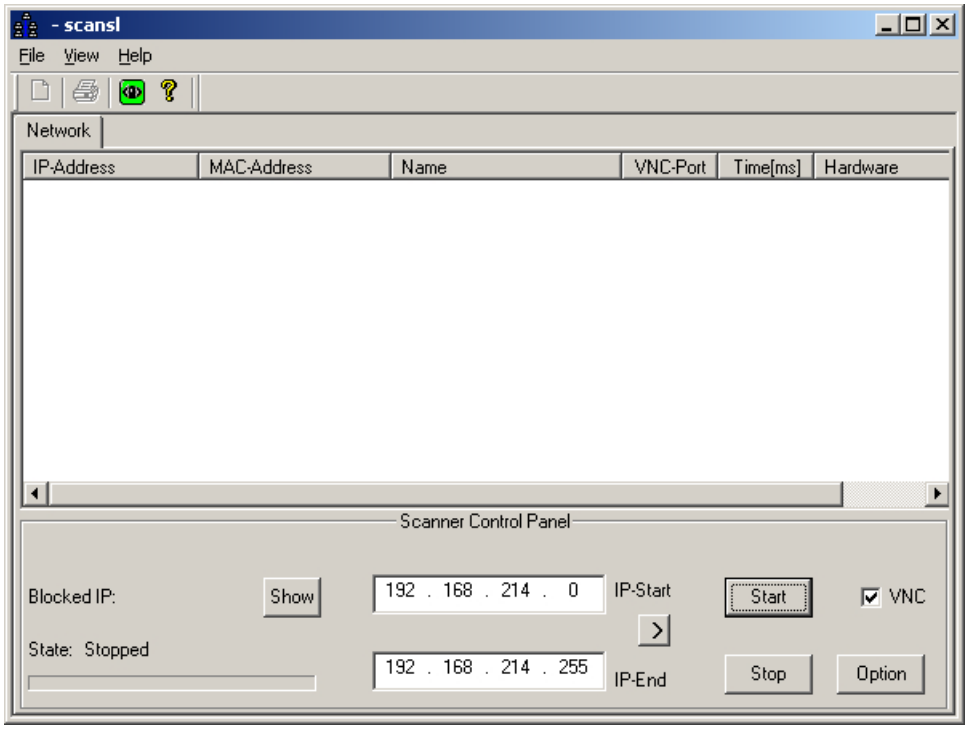



Figure 5-3 VNC scanner

2. "IP start / IP end": Set the IP-address browsing range.
   Eight ranges can be saved. Click on button " **>**" to switch to the next range.

1. "VNC" option:
   - ON: only search for VNC servers.
   - OFF: search all the stations in the network.
2. Start the search with the "Start" button.
   The following station parameters will be listed:
   - IP address
   - MAC address
   - Name in the network
   - VNC port
   - Response time of the station in milliseconds
   - Device type

   **Note**

   **local device #**

   When the devices are listed, an IP address will be identified with "#" to indicate that this is the local device on which the VNC scanner will be used.

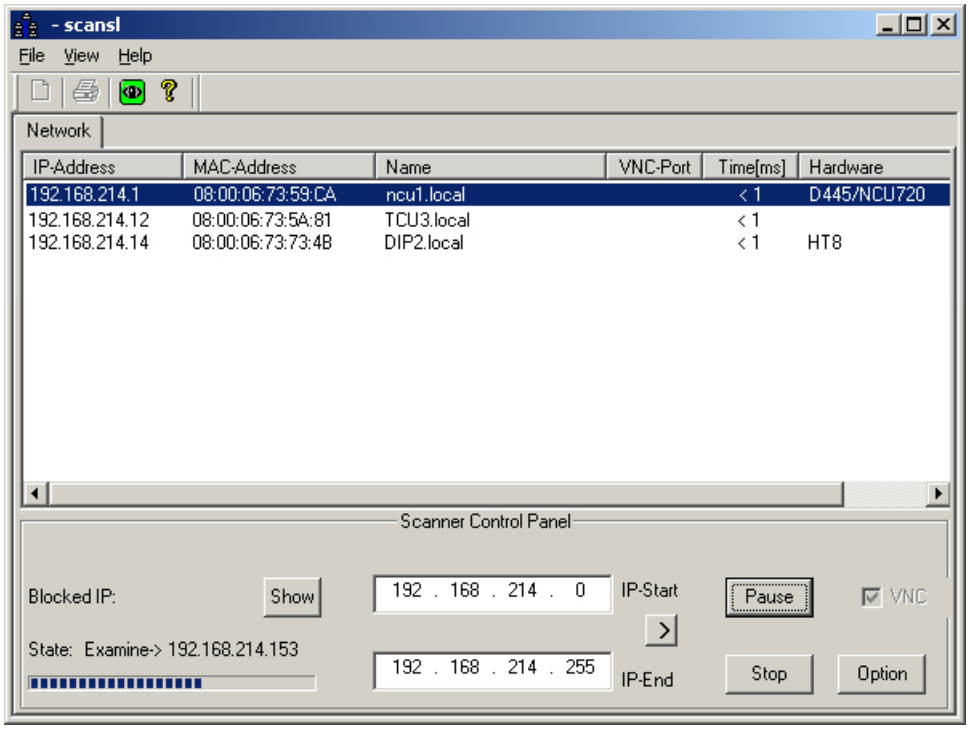



Figure 5-4 VNC scanner: List of stations

3. By right-clicking a network station from the list, you can execute the following actions via a pop-up menu:
   - "VNC": Connecting directly to the VNC server and, if applicable, operating the HMI.
   - "VNC → Auto reconnect": Corresponds to the option "Automatic reconnect if the server closes the connection" in order to restore the connection to the selected network station.

– Ping: Opens a command shell and executes the "ping" command to the selected network station.

**Note**

**Operation via a VNC viewer:**

A system can only be monitored from another station via an external VNC viewer (default).

To view the system from another station, this function must be enabled by the system. You can find the settings for this in the "System Network Center" or directly in tcu.ini in the [VNCViewer] section.

Should you wish to access a PCU via the company network using the VNC scanner or VNC viewer, port 5900 must be added to the PCU under "Control Panel" → "Windows Firewall" → "Exceptions".

**Reference:** Operator Components and Networking Manual

## Additional Options



1. Use this button to start the VNC viewer.
2. Use the "Option" button to set parameters for the search.

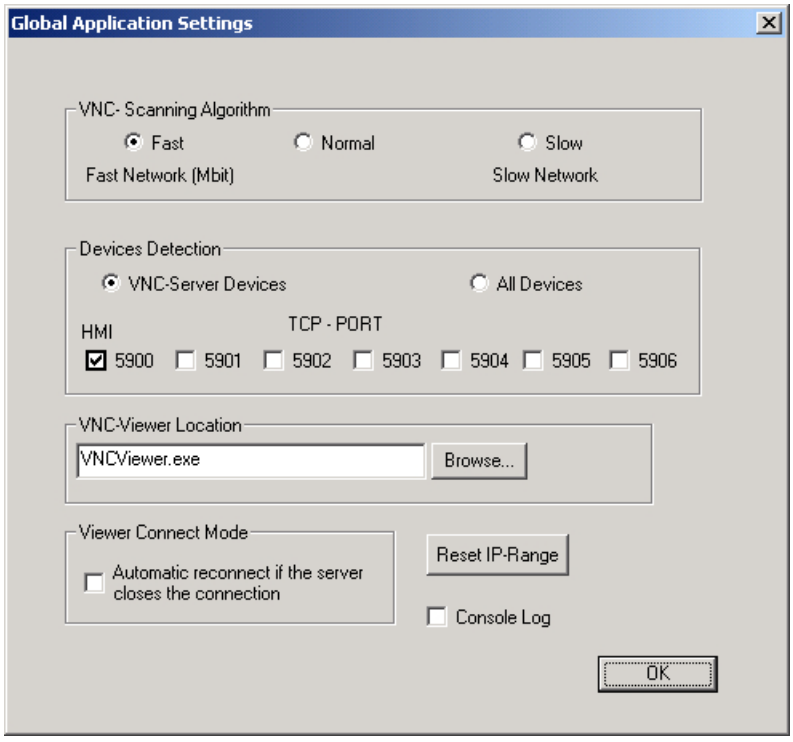



Figure 5-5 VNC scanner: Set search parameters

- VNC scanning algorithm:
  Default is "Fast", i.e. device feedback within approximately 100 ms
- Devices detection: (corresponds to the "VNC" option, see point 3).
  - "VNC server devices": only search for VNC servers.
  - "All devices": search all the stations in the network.
- VNC viewer location:
  Link to the VNC viewer (open source program: already pre-installed on the PCU; on a ServicePC this link must be specified).
- Viewer default connection mode:
  "Automatic reconnect if the server closes the connection" (default: OFF)
  - OFF: One attempt is made to establish a connection.
  - ON (loop mode): For situations where access will be via Internet, for example for service purposes, then the system will wait longer and try again after several milliseconds to establish a connection to this station.
- "Reset IP range": All IP address ranges are reset.
- Console log: Opening a console to output a protocol, e.g. for error analysis (default: OFF).

## Options for the VNC viewer

| NOTICE |
| --- |
| **Do not change defaults!**<br>To ensure that the VNC viewer works properly, the following options may not be changed. |

After starting the VNC viewer, the following dialog opens:

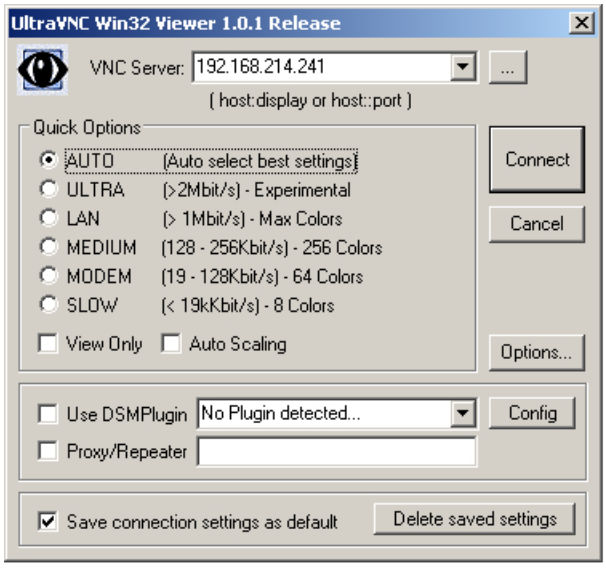



Figure 5-6 Default: UltraVNC

After clicking the "Options ..." button, the following dialog opens:

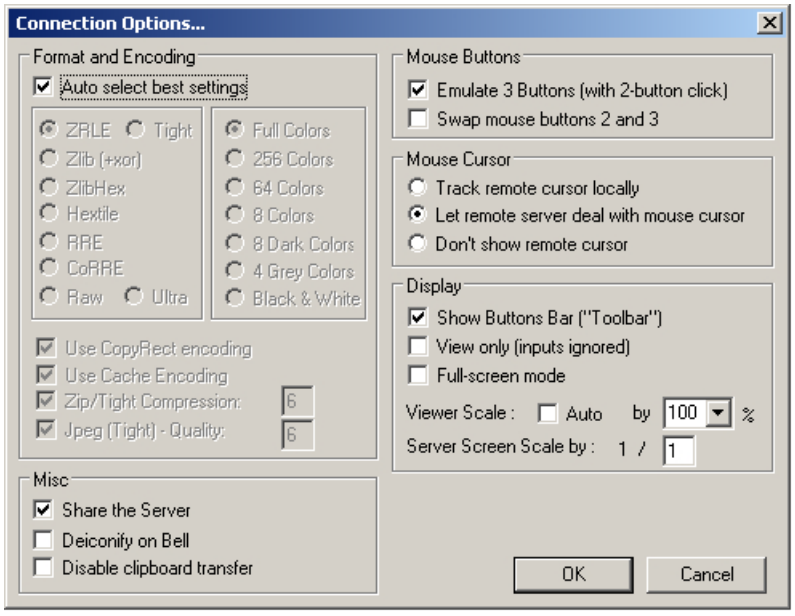



Figure 5-7 Default: Connection Options ...

# List of Abbreviations

# A

## A.1 Abbreviations

| | |
|---|---|
| CF | CompactFlash card: Memory card |
| CFS | Cluster File System |
| DCK | Direct Control Keys: Direct control keys |
| DCP | Discovery and Basic Configuration Protocol |
| DHCP | Dynamic Host Configuration Protocol: Dynamic assignment of an IP address and other configuration parameters on a computer in a network |
| DNS | Domain Name System: Conversion of domain names into IP addresses |
| EBS | Emergency Boot System |
| EKS | Electronic Key System: System to check the identity of a user (authentication system) |
| EUNA | End User Notification Administration |
| HMI | Human Machine Interface: Operator interface |
| IRT | Isochronous Realtime (Ethernet) |
| LLDP | Link Layer Discovery Protocol: multi-vendor Layer 2 Protocol defined in accordance with the IEEE-802.1AB standard, allows information to be exchanged between devices. |
| MAC | Media Access Control: The MAC address is a 48-bit Ethernet ID. |
| MCP | Machine Control Panel: Machine control panel |
| MPI | Multi-Point Interface: Multiple interface |
| MUI | Multilanguage User Interface |
| NCK | Numerical Control Kernel: NC kernel with block preparation, travel range, etc. |
| NCU | Numerical Control Unit: NCK hardware unit |
| NRT | Non-Realtime (Ethernet) |
| NTFS | New Technology File System |
| NTP | Network Time Protocol: Standard for synchronizing clocks in the entire network |
| NTPD | NTP Daemon:<br>Utility that runs in the background and does not have to be started by the user. |
| PCU | PC Unit: Computer unit |
| PDEV | Physical device |
| PG | Programming device |
| PLC | Programmable Logic Control: Programmable logic controller |
| RAM | Random Access Memory: Program memory which can be read and written into |
| RDY | Ready: The system is ready to operate. |
| SNMP | Simple Network Management Protocol (network protocol for monitoring and controlling network elements such as routers, servers, switches, and printers from a central station). |
| SSD | Solid State Drive |
| TCU | Thin Client Unit |
| TFTP | Trivial File Transfer Protocol: Very simple data transmission protocol |
| UDP | User Datagram Protocol: NTP is mostly processed via UDP. |
| USB | Universal Serial Bus |

| | |
|---|---|
| UPS | Uninterruptible power supply |
| UTC | Universal Time, Coordinated: Coordinated universal time |
| VNC | Virtual Network Computing |

# Index


## 7

7-segment display, 112

## A

Activating the firewall, 8

## B

Backup time, 62
BIOS
  Changing default setting, 31, 37
  Start setup, 28, 33
  System parameters, 29, 34
Boot screen
  OEM-specific, 47
  Storage location, 47
Boot stick, 96

## C

Changing computer name, 40
Communication
  MPI, 39
  S7DOS, 39
Configuration file
  basesys.ini, 100
  ghost.ini, 65
  servicesystem.ini, 65, 66, 67
Configuring the user interface, 45

## D

Dialog box
  ServiceCenter Backup/Restore, 65, 66, 67
Disk image
  Redo, 86, 90
  Troubleshooting, 87, 92
Domain, 42
Driver
  S7DOS, 39

## E

Emergency Boot System, 96
Enabling the log, 115
Error log, 115

## G

Global Settings, 10

## H

Hard disk
  Replacing, 109

## M

Mains-fed operation, 70
Multilingual User Interface (MUI), 74

## N

Network
  Adapting settings, 66, 67
Network release
  Drive of the PCU 50.5, 106
  Driven on PG/PC, 103
NTFS file system, 6

## P

Partitioning the hard disk, 6
Password
  For network drive, 93
PCU
  Checking the hard disk, 111
  Delivery condition, 5
  Name (default setting), 8
  Operating system, 7
PCU operating system, 7
Port 80 display, (See 7-segment display)
powerline, 39

## R

Recovery Media, 74
RESOLUTION (tcu.ini), 25

## S

Screen resolution, 25
Select language (MUI), 45
Service Desktop
    Background, 48
    install, 70
Service switch, 114
Service system for PCU, 96
Shutdown, 113
SIMATIC STEP 7
    Installing, 73
Software
    Components, 5
    install, 69
SSD
    Replacing, 109
Starting OEM programs, 54
System boot, 112

## T

Troubleshooting
    Restore failure, 87, 92

## U

USB FlashDrive, 96
User
    Change the name, 18
    Default, 9
    Delete, 18
    Inheritance, 16
User group, 9
User type, 9

## V

Version software components, 5
VNC scanner, 116

## W

Windows language (MUI), 45

SIEMENS

# SINUMERIK

# SINUMERIK 840D sl
# PCU Base Software (IM10)

Commissioning Manual


Delivery condition 1

First commissioning of the PCU 50.5 2

First commissioning of SIMATIC IPCs 3

Configuration of the system 4

Installing and configuring updates and automation software 5

Backing up and restoring data 6

Service and diagnostics 7


Valid for:

SINUMERIK PCU Base Software 50.5 W7/W10 V12.0
SINUMERIK PCU Base Software IPC V12.0


08/2018
6FC5397-1DP40-6BA2

## 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

Document order number: 6FC5397-1DP40-6BA2
℗ 08/2018 Subject to change

Copyright © Siemens AG 2013 - 2018.
All rights reserved

# Table of contents


1 Delivery condition....7

1.1 PCU Base Software Windows 7/10....7
1.1.1 Overview....7
1.1.2 Supplied software....9

1.2 Hardware configuration of PCU 50.5....10
1.2.1 Network settings....10
1.2.2 Partitioning of the SSD....10

1.3 Hardware configuration of SIMATIC IPC....12
1.3.1 Network settings....12
1.3.2 Partitioning of the SSD....12

1.4 Directory structure and file conventions....14

2 First commissioning of the PCU 50.5....15

2.1 Overview....15

2.2 Backing up and restoring the delivery condition....18

2.3 Install the PCU Base Software attended....19

2.4 Unattended installation of the PCU Base Software....20
2.4.1 Configuring an answer file....20
2.4.2 Installing the PCU Base Software attended....23

3 First commissioning of SIMATIC IPCs....25

3.1 Overview....25

3.2 Adapting the installation settings....26

3.3 Installing the PCU-Basesoftware....27

4 Configuration of the system....29

4.1 Overview....29

4.2 Managing user accounts....30

4.3 Changing the name of the PC system....31

4.4 Setting the IP address and domain....32

4.5 Configuring USB interfaces....33

4.6 Configuring network access on the USB data storage medium....34

4.7 Configuring the keyboard layout....35

4.8 Setting up an external screen....36

4.9 Setting the screen resolution....37

4.10 Set the resolution in the tcu.ini....38

4.11 Setting the color depth in the tcu.ini file....42

4.12 Activating/deactivating the multitouch operation of TCUs....43
4.13 Calibrating the touch operation of a screen....44
4.14 Configuring virtual direct control keys (VDCK)....45
4.14.1 General information....45
4.14.2 Creating a configuration file....46
4.14.3 Adapting the layout of the virtual direct control keys....47
4.14.4 Default setting for the direct control key layout....48
4.14.5 Reference to the vdck.ini configuration file....48
4.15 Configuring the SITOP UPS module for use with the PCU Base Software....51
4.15.1 Overview of SITOP UPS....51
4.15.2 SITOP modules for the PCU 50.5....52
4.15.3 Configuring SITOP software for PCU Base Software....53
4.15.3.1 Configuring SITOP software V3.x (USB)....53
4.15.3.2 Configuring SITOP UPS Manager (Ethernet)....58
4.15.4 Hardware configuration of the SITOP UPS module....59
4.16 Configuration of the Service Center....62
4.16.1 Overview....62
4.16.2 Configuration of the network adapter....63
4.16.3 Configuration of the host....64
4.17 Adapting the firewall settings....66
4.17.1 Overview....66
4.17.2 Factory setting of the Windows Firewall on the Siemens PC system....67
4.17.3 General information on the operating principle of the Windows Firewall....68
4.17.4 General information on the settings recommended for Windows....69
4.17.5 Configuration by means of a prompt or script/batch file....69
4.17.6 Enabling SNMP communication....70
4.17.7 Activate the remote access to the PC system....72
4.17.7.1 Overview....72
4.17.7.2 Via the Control Panel....72
4.17.7.3 By prompt, script or batch file....73
4.17.8 Activation/deactivation of the file and printer release....75
4.17.8.1 Overview....75
4.17.8.2 General information....76
4.17.8.3 General activation for all network profiles....77
4.17.8.4 Activation for a specific network profile....80
4.17.8.5 Activation for a specific connection....85
4.17.9 Activation of the ping execution (ICMP)....87
4.17.9.1 Via the Control Panel....87
4.17.9.2 By prompt, script or batch file....88
4.17.10 Saving and restoring firewall settings....89
4.17.10.1 Overview....89
4.17.10.2 Backup of firewall settings....90
4.17.10.3 Restoring of firewall settings....91
**5 Installing and configuring updates and automation software....95**
5.1 Overview....95
5.2 Patch management and security updates....96
5.3 Configuring saving Windows log files....97

5.4 Setting up SINUMERIK Operate for autologon mode....98
5.5 Modifying the user account for autologon mode....100
5.6 Configuring a key filter for HMI software....102
5.7 Installing STEP 7....106
5.8 Linking the HMI software with STEP 7....107
5.9 PCU Installer....108
5.9.1 Overview....108
5.9.2 Activating the PCU Installer....108
5.9.2.1 Overview....108
5.9.2.2 Activating via a configuration template....109
5.9.3 Deactivating the PCU Installer....110
5.9.4 Copying and adapting the configuration file....111
5.9.5 Reference to the PCUInst.ini configuration file....112
5.9.6 Example configuration: Installation behavior as for PCU base software Windows XP....116
5.9.7 Installing software with the PCU Installer....116
5.9.8 Procedure example of an installation with the PCU Installer....117
5.9.9 Log files (.log)....118
5.10 Migration....119
**6 Backing up and restoring data....121**
6.1 Overview....121
6.2 Starting the Service Center....122
6.3 Functions of the Service Center....123
6.4 Network settings in the Service Center....125
6.5 Create a disk image of the SSD....128
6.6 Restore a disk image of the SSD....130
6.7 Create a disk image of a partition....132
6.8 Restore a disk image of a partition....134
6.9 Network settings in the Service Center....137
6.10 Starting Symantec Ghost directly....139
6.11 Bootable USB flash drive....140
6.12 Operating a service PC/PC in the network....141
6.12.1 Overview....141
6.12.2 Connection options in the system network....141
6.12.3 Configuring routing in the network....144
6.12.4 Configure the network settings of the PG/PC....145
6.12.5 Release directory of the PC/PC in the network....148
6.12.6 Creating a shortcut to the network drive....151
6.13 Starting up the replacement SSD....153
**7 Service and diagnostics....155**
7.1 PCU Hardware Diagnostics....155

7.2 Evaluation of the diagnostic displays H1/H2....156
7.3 Setting of the operating mode during run-up....158
7.3.1 Service switch (PCU 50.5)....158
7.3.2 Possible operating modes....159
7.3.3 Software-side setting (SIMATIC IPC)....159
7.4 Switching to the Service Desktop during autostart / autologon operation....160
7.5 Configuring the SINUMERIK power up screen....162
7.6 Remote access....165
7.6.1 Overview....165
7.6.2 Searching for devices in the system network....165
7.6.3 Display of accessible stations in SINUMERIK Operate....165
7.6.4 Remote access to the PCU for operation and maintenance....166
7.6.5 Setting up and using SSH....169
7.6.6 Encryption via SSH protocol....170
**Index....171**

# Delivery condition

# 1

## 1.1 PCU Base Software Windows 7/10

### 1.1.1 Overview

The **PCU Base Software 50.5W7/W10/IPC** forms the basis for operating SINUMERIK software (e.g. SINUMERIK Operate) on your SIEMENS PC system:

- **SINUMERIK PCU 50.5**

  The PCU Base Software is included in the PCU 50.5 scope of delivery and preinstalled. However, for security reasons, the final installation and setup of user accounts must be performed during the first startup.

  An example flow diagram for initial startup is provided as an overview in Chapter Overview (Page 15). This overview assumes that the PCU 50.5 is already installed and networked.

  Information about the hardware configuration, installation, and networking of the PCU 50.5 is provided in the *SINUMERIK 840D sl Operator Components and Networking Manual*, Chapter *Operator panels*.

- **SIMATIC IPC (only in the following configurations)**

  The SIMATIC IPC627D, IPC427D, IPC477D, IPC427E and IPC477E industrial PCs are compatible with the PCU Base Software in the following configurations:
  - SIMATIC IPC627D configurations with the following article numbers:
    - 6AG4131-2GM30-0BA6
    - 6AG4131-2GM51-0BX6
  - SIMATIC IPC427D configurations with the following article numbers:
    - 6AG4140-0.C07-0HA0
    - 6AG4140-6.H17-0HA0
  - SIMATIC IPC427E configurations with the following article numbers:
    - 6AG4141-1AA14-0FA0
    - 6AG4141-5AB14-0FA0
  - SIMATIC IPC477D configurations with the following article number:
    - 6AV7240-0HC07-0HA0
  - SIMATIC IPC427E configurations with the following article numbers:
    - 6AG4141-1AA14-0FA0
    - 6AG4141-5AB14-0FA0
  - SIMATIC IPC477E configurations with the following article numbers:
    - 6AV7241-3YA04-0FA0
    - 6AV7241-5SB04-0FA0

The following SIMATIC panels can be operated on IPCs with PCU Base Software:

- SIMATIC IFP 2200 MT: 6AV7863-4MA00-0AAx
- SIMATIC IFP 2200 MT EXT: 6AV7863-4MB10-0AAx
- SIMATIC IFP 2200 MT: 6AV7466-8MA00-0MT0
- SIMATIC IFP 2200 MT EXT: 6AV7466-8MA10-0AX0
- SIMATIC IFP 1900 MT: 6AV7863-3MA00-0AAx
- SIMATIC IFP 1900 MT EXT: 6AV7863-6MA10-1AA0
- SIMATIC IFP 1900 MT EXT Neutral: 6AV7863-6MA10-1NA0
- SIMATIC IFP 1900 MT OEM: 6AV7456-7MA00-0GS0
- SIMATIC IFP 1500 MT EXT: 6AV7863-5MA10-1AA0
- SIMATIC IFP 1500 MT EXT Neutral: 6AV7863-5MA10-1NA0

The PCU Base Software is Windows 10-compatible with the SIMATIC IPC427E and IPC477E industrial PCs in the following configurations:

- SIMATIC IPC427E configurations with the following article numbers:
  - 6AG4141-1AA17-0FA0
  - 6AG4141-5AB17-0FA0
- SIMATIC IPC477E configurations with the following article numbers:
  - 6AV7241-3YA07-0FA0
  - 6AV7241-1WA07-0FA0
  - 6AV7241-5SB07-0FA0
  - 6AV7241-3XB07-0FA0

The following SIMATIC panels can be operated on Windows 10 IPCs with PCU Base Software (connection via Ethernet is not supported for IFP ETH panels):

- SIMATIC IFP 2200 MT: 6AV7466-8MA00-0MT0
  SIMATIC IFP 2200 MT EXT: 6AV7466-8MA10-0AX0
  SIMATIC IFP 1900 MT EXT: 6AV7863-6MA10-1AA0
  SIMATIC IFP 1900 MT EXT Neutral: 6AV7863-6MA10-1NA0
  SIMATIC IFP 1500 MT EXT: 6AV7863-5MA10-1AA0
  SIMATIC IFP 1500 MT EXT Neutral: 6AV7863-5MA10-1NA0

For these SIMATIC IPC configurations, you can order the SINUMERIK PCU Base Software/ IPC separately under the following article number:

- 6FC5800-0AP86-0YB0

Other or unnamed SIMATIC IPC configurations, screens or PC systems are not compatible with the PCU Base Software.

You can find an example workflow for commissioning and configuration under Overview (Page 25).

You can find all of the information on the hardware in the operating instructions for the corresponding SIMATIC IPC:

**Note**

**Do not switch off SIEMENS PC systems with Windows on the hardware!**

Loss of data may result on Windows-based systems if the system is not shut down properly before it is disconnected from the power source. For technical reasons, data is still being written to the SSD shortly after shutdown.

To avoid loss of data, use a SITOP UPS module.

See Configuring the SITOP UPS module for use with the PCU Base Software (Page 51).

## 1.1.2 Supplied software

### Overview

The SINUMERIK PCU Base Software contains the following software and tools:

| Software | Function |
|---|---|
| Microsoft Windows 7/10 | Service Desktop |
| Microsoft Windows PE | Service Center |
| Symantec Ghost | Creating and restoring disk images |
| SIEMENS PCU Installer | Installing software and updates |
| VNC Viewer | Remote access to other devices from the PCU |
| PuTTy | Remote access to other devices from the PCU |
| PuTTy Key Generator | Generating an SSH key pair |
| Various configuration templates and scripts in the directory: `C:\ProgramData\Siemens\MotionControl\SIEMENS` | Templates for simplifying configuration |

**Note**

**Acknowledgements**

- This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)
- This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)

# 1.2 Hardware configuration of PCU 50.5

## 1.2.1 Network settings

The PCU 50.5 has two Ethernet interfaces with default settings suitable for connection to SINUMERIK solution line:

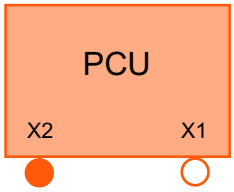



Figure 1-1 Ethernet interfaces of the PCU

| Hardware designations | Designation in PCU Base Software[1] | Configuration |
|---|---|---|
| X1 | Local Area Connection 2 | Preset as a DHCP client for connection to a **company network**.<br>The IP address is dynamically fetched from the DHCP server at X1. |
| X2 | Local Area Connection | Preset as a SINUMERIK DHCP server for connection to a **system network**.<br>A fixed IP address is preset at X2: If 192.168.214.241 is free in the system network, it is set.<br>Otherwise you can view the allocated fixed IP address in the Control Panel. |

[1] The network designation may deviate depending on the system.

### Further information

- *Manual SINUMERIK 840D sl Operator Components and Networking*
- Setting the IP address and domain (Page 32)
- Adapting the firewall settings (Page 66)
- Remote access (Page 165)
- Network settings in the Service Center (Page 125)

## 1.2.2 Partitioning of the SSD

The PCU 50.5 has a SSD with 80 GB storage capacity. The SSD is already partitioned and pre-assigned by the PCU base software as follows:

| **Name** | **System (C:)** | **Data (D:)** |
|---|---|---|
| **File system** | NTFS | NTFS |

| | | |
|---|---|---|
| **Storage capacity** | 40 GB | Approx. 40 GB |
| **Free capacity** | 20 GB | Approx. 40 GB |
| **Already occupied by** | • Microsoft Windows 7<br>• PCU Base Software incl. various applications and configuration templates (see Chapter Supplied software (Page 9)) | - |
| **Application** | Installing user programs | • Customer-specific data<br>• Backup images<br>• Setup packages |

# 1.3 Hardware configuration of SIMATIC IPC

## 1.3.1 Network settings

The supported SIMATIC IPC have two Ethernet interfaces which are set by the PCU Base Software for connection to the SINUMERIK solution line.

Table 1-1 SIMATIC

| Hardware designations | Designation in PCU Base Software[1] | Configuration |
|---|---|---|
| X1 | Local Area Connection 2 | Preset as a DHCP client for connection to a **company network**.<br>The IP address is dynamically fetched from the DHCP server at X1. |
| X2 | Local Area Connection | Preset as a SINUMERIK DHCP server for connection to a **system network**.<br>A fixed IP address is preset at X2: If 192.168.214.241 is free in the system network, it is set.<br>Otherwise you can view the allocated fixed IP address in the Control Panel. |

[1] The network designation may deviate depending on the system.

### Further information

- Operating instructions of the SIMATIC IPC
- Setting the IP address and domain (Page 32)
- Adapting the firewall settings (Page 66)
- Remote access (Page 165)
- Network settings in the Service Center (Page 125)

## 1.3.2 Partitioning of the SSD

The partitioning of the SSD of your SIMATIC IPC627D or IPC427D is not changed by the installation of the PCU-Basesoftware. Partition C: must have a minimum size of 30 GB to be able to install the PCU-Basesoftware.

| Name of the partition | System (C:) | Data (D:) |
|---|---|---|
| File system | NTFS | NTFS |
| Memory used by the system | ~ 20 GB | ~ 0 GB |

| | | |
|---|---|---|
| **Already occupied by** | • Microsoft Windows 7<br>• PCU-Basesoftware incl. various applications and configuration templates (see Chapter Supplied software (Page 9)) | - |
| **Application** | Installing user programs | • Customer-specific data<br>• Backup images<br>• Setup packages |

# 1.4 Directory structure and file conventions

### Templates for configuration files

To assist you with the parameterization of the system, templates for various configuration files are provided on the Siemens PC system. The templates are organized into various directories, which you will find below the following directory:

- `C:\ProgramData\Siemens\MotionControl\siemens`

These templates are already fully functional when supplied and will be used automatically if you do not create your own configuration files based on these templates.

---

**Note**

**Do not modify files in the template directory `siemens`!**

The `siemens` directory contains only files with the factory settings that can be copied and used as templates.

Never overwrite these files. Instead copy them into another directory (see Section *Your settings*).

---

### Directory hidden in delivery condition

For security reasons, the `C:\ProgramData` directory is hidden in the delivery condition.

To view the directory, proceed in one of the following ways:

- To switch quickly to a hidden directory, type the address in the Windows Explorer.
- To show all hidden directories by default, change the setting in the Control Panel under "Appearance and Personalization > Folder Options > View > Advanced Settings".
- To show only one directory by default, first make the setting to show all hidden directories, then activate the attribute "Hidden" in the properties of the directory and finally again make the setting to not show all hidden directories.

### Your settings

Several directories are available into which you can copy and adapt configuration files in order to make your own settings. In decreasing order of priority, these are:

- `C:\ProgramData\Siemens\MotionControl\`**user**
- `C:\ProgramData\Siemens\MotionControl\`**oem**
- `C:\ProgramData\Siemens\MotionControl\`**addon**

The settings in a configuration file in a directory with higher priority always replace those from one of lower priority. Template directory `siemens` has the lowest priority.

To define your own settings, therefore, copy a setting from the relevant template file into a directory with higher priority, e.g. below directory `user`.

# First commissioning of the PCU 50.5

# 2

## 2.1 Overview

Once you have installed and networked the PCU 50.5, you must set up a Windows administrator account before the PCU 50.5 can be accessed from a remote TCU. In headless mode, therefore make the configuration in a response file (unattend.xml) before switching on the PCU 50.5 for the first time.

The diagram shows a sample workflow for the first commissioning.  Further information is provided in the accompanying legend tables.

---

**Note**

**Do not switch off the PCU 50.5 Windows 7 hardware!**

Loss of data may result on Windows-based systems if the system is not shut down properly before it is disconnected from the power source. For technical reasons, data is still being written to the SSD shortly after shutdown.

To avoid loss of data, use a SITOP UPS module.

See Configuring the SITOP UPS module for use with the PCU Base Software (Page 51).

---

## Example workflow

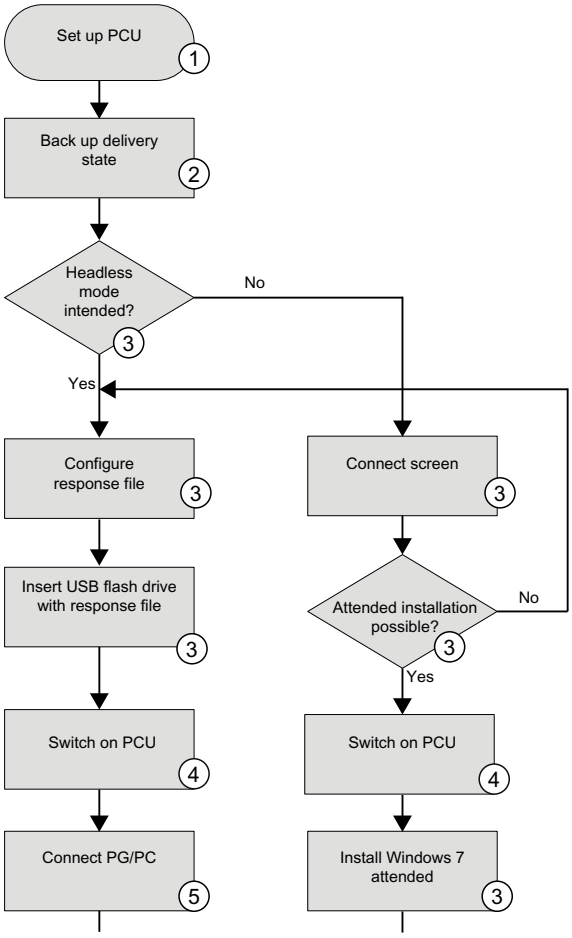



Figure 2-1 Example sequence of first commissioning

| No. | Step | Description |
|---|---|---|
| 1 | Set up PCU | Information about installation and mounting of the PCU 50.5 is provided in the *SINUMERIK 840D sl Operator Components and Networking Manual*, Chapter *Operator panels*. |
| 2 | Back up delivery state | Before starting the actual first commissioning, you can back up the delivery condition of the Windows 7 PCU base software as disk image (Page 18). |
| 3 | • Headless mode intended?<br>– Configure response file<br>– Insert USB flash drive with response file<br>• Attended installation possible?<br>– Connect screen<br>– Install Windows 7 attended | As soon as you start with the first commissioning of the PCU base software, you must decide on an installation type:<br>• Configure a response file and complete installation and user settings unattended (Page 20).<br>• Connect screen (Page 36) and complete installation attended (Page 19), at the same time set up users. |
| 4 | Switch on PCU | Do not switch on the PCU without the response file (Page 20) if you want the installation to be performed unattended. |
| 5 | Connect PG/PC | To be able to access a directory of a PG/PC in the network, for example, connect the PG/PC in the network (Page 141). |

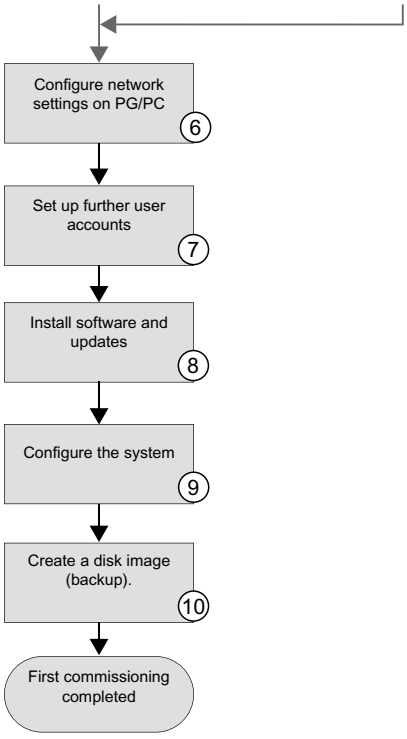



Figure 2-2 Example sequence of first commissioning (continued)

| No. | Step | Description |
|---|---|---|
| 6 | Configure network settings on PG/PC | Once you have networked the PCU, a PG/PC and any other devices with one another, configure the network settings of your PG/PC (Page 141). |
| 7 | Set up further user accounts | Before you install software, at least one user account with limited rights must be set up (Page 30). |
| 8 | Install software and updates | Install necessary automation software and Windows updates (Page 95). |
| 9 | Configure the system | Configure the installed software (Page 95), Windows 7 and other system properties (Page 29) and, if necessary, the SITOP UPS module (Page 51). |
| 10 | Create a disk image (backup). | Create a disk image of the SSD as a backup (Page 121) to be able to return to this version in a maintenance situation. |

## 2.2 Backing up and restoring the delivery condition

Before starting the actual first commissioning, you can backup the delivery condition of the PCU base software.

### Requirement

The first commissioning of the PCU 50.5 (with "unattend.xml" response file or attended) has not been performed yet.

### Procedure

To back up the delivery condition of the PCU base software, proceed as follows:

1. Start the Service Center directly after switching on the PCU 50.5 the first time, e.g. via the Windows boot menu.
   Starting the Service Center (Page 122)
2. Back up the entire `C:` partition as disk image at another location, e.g. at `D:\DeliveryState_PCU-Basesoftware\`
   See also: Create a disk image of a partition (Page 132)

### Result

The delivery condition is stored as disk image at the selected location. If required, you can move this disk image to another location later.

First create a bootable USB flash drive (Page 140) with which you can restore this disk image (Page 134).

After you have restored the delivery condition of a PCU 50.5 with a disk image, the first commissioning starts at the initial startup with response file or attended.

### Further information

- Backing up and restoring data (Page 121)
- First commissioning of the PCU 50.5 (Page 15)

## 2.3 Install the PCU Base Software attended

If you have connected to the PCU 50.5 directly to the operator panel front or screen, you can complete installation of the PCU Base Software and Windows 7 attended. For this, user entries are required during installation in order to set up the Windows administrator account, for example.

Alternately, you can use unattended installation (Page 20) in which case you define the necessary user entries in an answer file (unattend.xml) before switching on the PCU 50.5.

### Precondition

You can connect an operator panel front or screen and keyboard directly to the PCU 50.5.

### Procedure

To perform attended installation of the PCU Base Software, proceed as follows:

1. Switch on the PCU 50.5.
   The installation wizard guides you through the necessary steps.
2. During installation, create a Windows administrator account. You can create additional user accounts in the Control Panel later. See also: Managing user accounts (Page 30).
3. Complete installation.

### Result

Installation of the PCU Base Software with Windows 7 is performed. After attended installation, you must activate the PCU Installer manually (Page 108).

## 2.4 Unattended installation of the PCU Base Software

You can also install the PCU Base Software unattended.

Use this procedure in the following cases:

- If you are operating a PCU as a headless system (without screen and input devices)
- If you want to start up several PCUs with identical settings

To start unattended installation (Page 23), during initial startup use a USB flash drive with answer file "unattend.xml" in its root directory. You preliminarily define the settings that you would normally have to make as user entries during installation in this answer file.

### See also

Microsoft TechNet - Methods for performing Windows setup (http://technet.microsoft.com/en-us/library/dd744269.aspx)

### 2.4.1 Configuring an answer file

#### Overview

The PCU-Basesoftware, Windows 7, is pre-installed on your PCU 50.5, but for security reasons, final installation and the setting up of user accounts must be performed during initial startup.

**Configuration using the unattend.xml response file**

A response file (`unattend.xml`) allows Microsoft Windows to be installed unattended. In this file, you define the settings in advance, which you must normally enter as user data during installation. Before you switch on the PCU for the first time, copy your adapted unattend.xml into the root directory of a USB flash drive and connect this to the PCU 50.5.

Unattended installation is required in the following cases:

- If you are operating a PCU as a headless system (without screen and input devices)
- If you want to start up several PCUs with identical settings

**Obligatory settings**

You preliminarily define the settings that you would normally have to make as user entries during installation in unattend.xml. You must enter at least the following information:

- Name of the administrator account
- Password of the administrator account
- Password hint
- Confirmation of the EULA

### Execute the command line commands unattended

The `unattend.xml` response file with the element /FirstLogonCommands/ **SynchronousCommand** provides you with the option to automatically execute commands unattended when Windows first runs up. However, to do this, you must also activate the autologon mode with the administrator account.

You can define as many FirstLogonCommands as required. You define the sequence in which the commands are executed in the element /FirstLogonCommands/SynchronousCommand/ **Order**

In the response file used as example, two different command line commands are executed with FirstLogonCommands/**SynchronousCommand**:

- Directory `D:\Install` released for the administrator account
- Activate the PCU Installer using the response file

### Sample response file

Because settings are always user-specific, only a sample response file can be provided as an example:

- Sample answer file for PCU 50.5 Windows 7 in the Service&Support Portal (http://support.automation.siemens.com/WW/view/en/69154062)

Adapt this file to your requirements.

Table 2-1 Mandatory and recommended settings in response file "unattend.xml"

| Path in the XML data structure | Purpose | Example |
|---|---|---|
| unattend/<br>settings/<br>component/<br>**AutoLogon** | **Modify the user account for the autologon mode (optional).**<br>If you want to automatically execute command line commands (`FirstLogonCommands`) after the unattended installation of Windows, activate the autologon mode in the `unattend.xml`.<br>Enter the password and user name of an administrator account that you also want to create with this response file.<br>`FirstLogonCommands` are only automatically executed when the administrator account is active. | `<AutoLogon>`<br>`<Password>`<br>`<Value>SUNRISE</Value>`<br>`</Password>`<br>`<Enabled>true</Enabled>`<br>`<Username>auduser</Username>`<br>`</AutoLogon>` |
| unattend/<br>settings/<br>component/<br>OOBE/<br>**ProtectYourPC** | **Settings for Windows update**<br>Set the value `1` to ensure that important and recommended Windows updates are automatically installed. | `<ProtectYourPC>1</ProtectYourPC>` |
| unattend/<br>settings/<br>component/<br>OOBE/<br>**NetworkLocation** | **Setting the network type and network identifier**<br>Set the value `Other`. | `<NetworkLocation>Other</NetworkLocation>` |

| **Path in the XML data structure** | **Purpose** | **Example** |
|---|---|---|
| unattend/<br>settings/<br>component/<br>OOBE/<br>**HideEULAPage** | **Confirm EULA (obligatory)**<br>Set the value to `true` to automatically confirm EULA. | `<HideEULAPage>true</HideEULAPage>` |
| unattend/<br>settings/<br>component/<br>UserAccounts/<br>LocalAccounts/<br>**LocalAccount** | **Create the administrator account (obligatory)**<br>Enter a user name in the `Name` element, and a secure password in the `Password/Value` element.<br>Specify the administrator status in the `Group` element.<br>The `DisplayName` and `Description` elements are optional.<br><br>In order to create additional user accounts, insert the complete `LocalAccount` section a second time and there, change the values of `Name`, `Password/Value` and, if required, `Group`. | `<LocalAccount wcm:action="add">`<br>`<Name>auduser</Name>`<br>`<DisplayName>auduser</DisplayName>`<br>`<Description>First admin account</Description>`<br>`<Group>Administrators</Group>`<br>`<Password>`<br>`<Value>SUNRISE</Value>`<br>`<PlainText>true</PlainText>`<br>`</Password>`<br>`</LocalAccount>` |
| unattend/<br>settings/<br>component/<br>FirstLogonCommands/<br>**SynchronousCommand** | **Activate PCU Installer (optional)**<br>In order that the PCU-Basesoftware component PCU Installer searches for setup files directly after logging on, add this command.<br>As result, a setting is made in the ini file of the PCU Installer.<br>So that you can store setup files in the `D:\Install` default installation directory, you must release this for the administrator account in the `unattend.xml` (see above).<br><br>For further information, see Chapter PCU Installer (Page 108). | `<SynchronousCommand wcm:action="add">`<br>`<CommandLine>"C:\Program Files (x86)\Siemens\MotionControl\Siemens\System\etc\write_ini_entry.exe" "C:\ProgramData\Siemens\MotionControl\user\System\etc\PCUInst.ini" processing_loginphase StartState activated ENH</CommandLine>`<br>`<Description>Set PCU Installer to start at user login</Description>`<br>`<Order>1</Order>`<br>`</SynchronousCommand>` |
| unattend/<br>settings/<br>component/<br>FirstLogonCommands/<br>**SynchronousCommand** | **Release the default installation directory (optional)**<br>Use this sample code to release directory D:\Install for the administrator code.<br>The sample code sets the access rights for `D:\Install` to `FULL` (read and write access) for the "auduser" user account. Instead of "auduser", enter the name of your administrator account. | `<SynchronousCommand wcm:action="add">`<br>`<CommandLine>net share Install=D:\Install /GRANT:auduser,FULL /REMARK:Installshare</CommandLine>`<br>`<Description>Share the local built-in installation folder</Description>`<br>`<Order>2</Order>`<br>`</SynchronousCommand>` |

Optionally, you can configure additional settings such as additional user accounts or network settings.

### Further information

You will find information about all options for adapting unattend.xml on the Microsoft website.

- Microsoft TechNet - Methods for performing Windows setup (http://technet.microsoft.com/en-us/library/dd744269.aspx)
- Microsoft TechNet - Unattended Windows Setup Reference (http://technet.microsoft.com/en-us/library/ff699026.aspx)

## 2.4.2 Installing the PCU Base Software attended

If you want to operate a PCU as a headless system or start up several PCUs with identical settings, use unattended installation.

### Requirement

- The PCU is in the delivery condition and switched off.
- In BIOS, the startup of the USB interfaces is deactivated.
  In the PCU50.5, it is deactivated via a factory setting.
- A USB flash drive is available (e.g. SIMATIC PC USB-FlashDrive).
  The USB flash drive must not use encryption, or it must be deactivated.
- The "unattend.xml" response file has been adjusted to your settings and at least one administrator account with password is defined.
  See: Configuring an answer file (Page 20).

### Procedure

Proceed as follows to start unattended installation:

1. Copy your adjusted "unattend.xml" response file directly into the root directory of the USB flash drive.
2. Connect the USB flash drive to the PCU.
3. Switch on the PCU.

### Result

The first startup is performed as an unattended installation, using the settings that you specified in the response file.

# First commissioning of SIMATIC IPCs 3

## 3.1 Overview

After you have completed the first commissioning of yoursupported SIMATIC IPC configuration (Page 7) with the PCU-Basesoftware, you can configure the SIMATIC IPC like a SINUMERIK PCU.

You can find an example workflow of the first commissioning and configuration in this tabular overview.

**Note**

**Do not switch off SIEMENS PC systems with Windows 7 on the hardware!**

Loss of data may result on Windows-based systems if the system is not shut down properly before it is disconnected from the power source. For technical reasons, data is still being written to the SSD shortly after shutdown.

To avoid loss of data, use a SITOP UPS module.

See Configuring the SITOP UPS module for use with the PCU Base Software (Page 51).

### Example workflow

| No. | Step | Description |
|---|---|---|
| 1 | Set up and commissioning of the PC system (according to the operating instructions) | Information on setting up and installing your SIEMENS PC system can be found in the corresponding hardware documentation: |
| 2 | Install the PCU-Basesoftware | If required, adapt the installation settings (Page 26) and install the PCU-Basesoftware (Page 27). |
| 3 | Connect the PG/PC | To be able to access a directory of a PG/PC in the network, for example, connect the PG/PC in the network (Page 141). |
| 4 | Configure network settings on PG/PC | Once you have networked the PC system, a PG/PC and any other devices with one another, configure the network settings of your PG/PC (Page 141). |
| 5 | Set up further user accounts | Before you install software, at least one user account with limited rights must be set up (Page 30). |
| 6 | Install software and updates | Install necessary automation software and Windows updates (Page 95). |
| 7 | Configure the system | Configure the installed software (Page 95), Windows 7 and other system characteristics (Page 29) and, if necessary, the SITOP UPS module (Page 51). |
| 8 | Create a disk image (backup) | Create a disk image of the SSD as a backup (Page 121) to be able to return to this version in a maintenance situation. |

## 3.2 Adapting the installation settings

The setup of the PCU Base Software also contains the setup.ini configuration file which can be parameterized before calling the setup. Note that some settings are only suitable for test purposes.

Before you adapt the settings in this file, you should copy the original file.

| Section | Setting | Description |
|---|---|---|
| [IPC] | FirewallSettings= | • 1 (default setting)<br>The Windows firewall settings are adapted appropriately for the company network by the PCU Base Software, as described in this documentation.<br>• 0<br>The current firewall settings are not changed by the setup. |
| | InstallWin7Updates= | • 1 (default setting)<br>During setup of the PCU Base Software, important Microsoft Windows updates are installed that are required for the use of the PCU Base Software. If these updates cannot be installed, the setup of the PCU Base Software is aborted.<br>• 0<br>The setup does not check whether the Microsoft Windows updates required for the PCU Base Software have been installed or can be installed. This setting may be required when new updates have already been installed on the PC system, for example. |
| | TestSystemEnglish= | • 1 (default setting)<br>The setup checks whether an English Windows version has been installed.<br>• 0<br>The setup does not check the Windows language version. In this case, there is an additional prerequisite that the names of the network adapter do not, for example, contain a hyphen.<br>Note that this setting is only suitable for test purposes. The PCU Base Software is only compatible with English Windows versions. |
| [SetupControl] | FinalReboot= | • 1 (default setting)<br>The PC system is restarted after installation.<br>• 0<br>The setup does not restart the system or call a dialog after the installation as to whether a restart is to be performed (supervised installation). |

## 3.3 Installing the PCU-Basesoftware

You can install the SINUMERIK PCU-Basesoftware/IPC on the supported SIMATIC IPC configurations (Page 7) (by means of an installation medium that can be ordered and licensed individually).

In the Industry Mall you can find the SINUMERIK PCU-Basesoftware/IPC under the following article number:

- 6FC5800-0AP86-0YB0

### Requirement

- You can connect an operator panel front or screen and keyboard directly to the PC system.
- The commissioning of the PC system is completed and an administrator account is set up.
- There is an English Windows installation on the PC system.
- The partition C: of the PC system has a minimum size of 30 GB.
- The installation medium is available on the PC system.

### Procedure

To install the PCU-Basesoftware on your PC system, proceed as follows:

1. Open Windows Explorer and switch to the installation medium of the PCU-Basesoftware.
2. Double-click the file "setup.exe".
3. Restart the PC system after you have installed the PCU-Basesoftware.

### Result

The SINUMERIK PCU-Basesoftware has been installed. During the installation, the previously existing settings on the SIEMENS PC system, e.g. user accounts, keyboard layout, date and time settings, were imported. Set up additional user accounts (Page 30) if necessary and set the keyboard layout to English (Page 35).

The PCU Installer is at first disabled after the installation. It must be enabled (Page 108) before software can be automatically installed.

# Configuration of the system

# 4

## 4.1 Overview

Once you have completed the first commissioning, you can adapt the configuration of the system:

- For security reasons, set up user accounts (Page 30) with limited rights, e.g. for auto logon mode.
- You can view or change (Page 31) the name of the PC system.
- The network settings (Page 32) in the delivery condition have been adapted for use with SINUMERIK solution line and usually do not have to be changed.
- You can deactivate the USB interfaces (Page 33) to prevent malicious software from entering the system network via this route.
- You can set up an external screen (Page 36) or set the screen resolution (Page 37) and color depth (Page 42).
- If you use a SITOP UPS module for the uninterruptible power supply or plan to use one, configure it for use with the PCU base software (Page 51).

Other configuration options (e.g. Desktop background) are available directly from the Service Desktop and can be implemented by the known Windows 7 methods:

- To call up the Windows help, click "Help and Support" in the Start menu.
- You will find help for Windows 7 in the Internet on the Microsoft website: Microsoft Windows support (http://windows.microsoft.com/en-US/windows/windows-help#windows=windows-7)

# 4.2 Managing user accounts

## Overview

During initial commissioning, you must create a local Windows administrator account and define a password.

You can set up and manage additional Windows user accounts in the Control Panel. To do that, log on with administrator rights.

---

**Note**

**Administrator account cannot be recovered**

If you have forgotten the password of the administrator account there is no way of recovering it!

However, after initial commissioning, you can create a file with which you can later assign a new password. To do that, in the Control Panel under user accounts use the "Create a password reset disc" command.

See also: Microsoft Windows support - creating a password reset disk (http://windows.microsoft.com/en-US/windows7/Create-a-password-reset-disk)

---

## Further information about user administration under Windows 7

You will find information about user administration under Windows 7 in the Windows help.

- To call up the Windows help, click "Help and Support" in the Start menu.
- You will find help for Windows 7 in the Internet on the Microsoft website: Microsoft Windows support (http://windows.microsoft.com/en-US/windows/windows-help#windows=windows-7)

## 4.3 Changing the name of the PC system

A unique computer name must be set after installation of the PCU base software. You can view and change it in the Control Panel of the Service Desktop.

You will find the computer name in the Control Panel under Category "System and Security", Section "Computer name, domain, and workgroup settings",

## 4.4 Setting the IP address and domain

The network settings in the delivery condition have been adapted for use with SINUMERIK solution line and usually do not have to be changed.

See also: Network settings (Page 10)

If you want to distinguish your network configuration from the factory settings, you can adapt them in the Windows Control Panel.

You will find these settings in the following menu:

- "Start > Control Panel > Network and Sharing Center > View network connections"

### Further information

You can adapt the network settings for backing up and restoring disk images in the configuration of the Service Center (Page 62).

You will find information about network settings under Windows 7 in the Windows help.

- To call up the Windows help, click "Help and Support" in the Start menu.
- You will find help for Windows 7 in the Internet on the Microsoft website: Microsoft Windows support (http://windows.microsoft.com/en-US/windows/windows-help#windows=windows-7)

# 4.5 Configuring USB interfaces

The USB interfaces are activated in the as-delivered state of the PC system.
To prevent malicious software from entering the control or the system network via the USB interfaces, you can deactivate the USB interfaces.

You can configure the USB interfaces on the Service Desktop via a command in the "Run" dialog box or at the prompt.

## Procedure

To activate or deactivate the USB interfaces, enter the relevant command on the Service Desktop in the "Run" dialog box or at the prompt:

Table 4-1 Disabling USB interfaces with the "sc_usb disable" command

| | |
|---|---|
| Factory setting: | The USB interfaces are enabled in the delivery condition. |
| Syntax: | sc_usb disable [-minutes] [all] [HOSTS...] |
| Description: | • Without a time indication [-minutes], a permanent disable is set. With time indication, the disable takes the time set in minutes.<br>• The disable may refer to all network nodes of the system network, or a list of host names or IP addresses may be specified to which the disable should apply. |

Table 4-2 Activating USB interfaces with the "sc_usb enable" command

| | |
|---|---|
| Factory setting: | The USB interfaces are enabled in the delivery condition. |
| Syntax: | sc_usb enable |
| Description: | With "sc_usb enable" USB storage units on permanently disabled USB interfaces or on certain host names or IP addresses are enabled again. |

## Further information

You can find general information about a prompt on the Microsoft website.

- Prompt: Frequently asked questions (windows.microsoft.com/en-us/windows/command-prompt-faq)

# 4.6 Configuring network access on the USB data storage medium

If you connect a removable data storage medium (e.g. a USB flash drive) to a USB interface of the PC system, a network drive is set up automatically. In this way, a removable data storage medium can be accessed from an NCU in the network. The file and the printer release is activated automatically by Windows.

If you do not require network access to USB removable data storage medium, you should deactivate the automatic network release.

You can find this setting in the `basesys.ini` at `EnableUSBShares`.

## Requirement

- Hidden directories are not hidden
- A copy of the basesys.ini configuration file is stored in the `user` directory: `C:\ProgramData\Siemens\MotionControl\`**user**`\System\etc\basesys.ini`
- The `basesys.ini` configuration file in the `user` directory is not write-protected

See: Directory structure and file conventions (Page 14)

## Procedure

To configure the automatic network release of USB data storage media of the PC system, proceed as follows:

1. Open the basesys.ini configuration file in the `C:\ProgramData\Siemens\MotionControl\`**user**`\System\etc\` directory.
2. Remove the semicolon in front of the EnableUSBShares line and set the required value.
   - `EnableUSBShares=0`
     The automatic network release of USB data storage media (and the associated file and printer release) is deactivated.
   - `EnableUSBShares=1` (default value)
     The automatic network release of USB data storage media is activated.
3. Save and close the file.

## 4.7 Configuring the keyboard layout

It is assumed that the English keyboard layout is used in SINUMERIK Operate and for the SINUMERIK operator panel fronts.

If a different keyboard layout is set in Windows, this should therefore be changed before commissioning the SIEMENS PC system or before using SINUMERIK Operate.

The setting of the keyboard layout as well as the display of the active keyboard layout is shown in the Windows task bar.

Further information is available at: Microsoft Windows support: Language bar (overview) (http://windows.microsoft.com/en-us/windows/language-bar-overview#1TC=windows-7)

# 4.8 Setting up an external screen

You can connect an external screen to the Siemens PC system. The procedure follows the usual procedure under Windows 7.

## Requirements

- The PC system is switched off.
- The external screen is connected to the DVI interface of the PC system (with an adapter, if necessary).

## Procedure

To use an external screen for the Service Desktop or service system, proceed as follows:

1. Let the PC system boot in the Service desktop.
2. Right-click on the Desktop, then select "Screen Resolution" from the shortcut menu.
3. In the "Screen Resolution" dialog box, select the screen and set the resolution:
   - If you are using an LCD or LED screen, you should ideally set the native resolution of your screen.
   - If you are using an older CRT screen, you can select any resolution.
4. Click "OK" to confirm the settings.

## Result

The external screen has been connected and can be used.

## 4.9 Setting the screen resolution

You can set the screen resolution directly in Windows 7 or use the PCU base software specific file tcu.ini
The settings in tcu.ini overwrite the Windows system settings. Compared with Windows, they offer additional functions for switching over the resolution of the PC system depending on, for example, the screen connected during run-up.

- Information about the configuration file tcu.ini is provided in the following chapter (Page 38).
- Information for setting the screen resolution under Windows 7 is provided on the Microsoft website: Changing the screen resolution (http://windows.microsoft.com/en-us/windows7/change-your-screen-resolution)

# 4.10 Set the resolution in the tcu.ini

## Overview

The system behavior during run-up for the screen resolution is set in the `tcu.ini` file in Section [VNCServer].

You will find a template of the `tcu.ini` under `C:\ProgramData\Siemens\Motion Control\siemens\System\etc\`
Do not overwrite this template, but rather create your own `tcu.ini` inside one of the user directories (see Chapter Directory structure and file conventions (Page 14)).

## Further information

- Chapter Directory structure and file conventions (Page 14).
- Operator Components Manual, Chapter "Networking"

## Setting the resolution during power up of the PC system

The following options are available for selection in the tcu.ini file in the [VNCServer] section (below the `# RESOLUTION` comment):

```
0 = SYSTEM
1 = AUTO_OP_1 (default)
2 = AUTO_OP_2
3 = AUTO_MON_1
4 = AUTO_MON_2
5 = 640x480
6 = 800x600
7 = 1024x768
8 = 1280x1024
9 = 800x480
10 = 1280x800
11 = 1366x768
12 = 1920x1080
```

The meaning of the settings are as follows:

Table 4-3 Settings in the configuration file `tcu.ini`

| Settings | Meaning |
|---|---|
| SYSTEM | The resolution is not specially set during run-up. i.e. the resolution last used in the system is active, e.g. the resolution which had been set manually in the Control Panel. |
| AUTO_OP_1 | Default setting:<br>During run-up, the resolution is automatically set ("PCU panel" has priority) in accordance with the following scenarios: |
| Example 1: | There is a PCU panel (irrespective of whether there is a PCU monitor and TCU panels):<br>[In active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is set to the max. resolution of the PCU panel (max. 1920x1080). |
| Example 2: | There is no PCU panel, however there is a PCU monitor (irrespective of whether there are any TCU panels):<br>[In active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is not specially set; i.e. the resolution last used in the system is active, e.g. the resolution which had been set manually in the Control Panel.<br>(Different to AUTO_OP_2 !) |
| AUTO_OP_2 | Like AUTO_OP_1, except:<br>Example 2: There is no PCU panel, however there is a PCU monitor (irrespective of whether there are any TCU panels):<br>[In active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is set to the max. resolution of the PCU monitor, reduced to the next lowest SINUMERIK resolution. The SINUMERIK resolutions are 640x480, 800x600, 1024x768, 1280x1024, 800x480, 1280x800, 1366x768 and 1920x1080.<br>**Example:** In the case of a PCU monitor with a max. resolution of 2048x1152, the SINUMERIK resolution setting is 1920x1080. |
| AUTO_MON_1 | During run-up, the resolution is automatically set ("PCU monitor" has priority) in accordance with the following scenarios: |

| Settings | Meaning |
|---|---|
| Example 1: | There is a PCU monitor (irrespective of whether there is a PCU panel and TCU panels):<br>[In active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is set to the max. resolution of the PCU monitor, reduced to the next lowest SINUMERIK resolution. The SINUMERIK resolutions are 640x480, 800x600, 1024x768, 1280x1024, 800x480, 1280x800, 1366x768 and 1920x1080.<br>**Example:** In the case of a PCU monitor with a max. resolution of 2048x1152, the SINUMERIK resolution setting is 1920x1080.<br>If there is a PCU panel, the display there is panned if the max. resolution of the PCU panel is lower than the max. resolution of the PCU monitor. |
| Example 2: | There is no PCU monitor, however there is a PCU panel (irrespective of whether there are any TCU panels):<br>[In active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is not specially set; i.e. the resolution last used in the system is active, e.g. the resolution which had been set manually in the Control Panel.<br>(Different to AUTO_MON_2 !) |
| Example 3: | There is no PCU monitor and no PCU panel (headless operation):<br>A) In active TCU mode:<br>a) Dynamic resolution switching is enabled (resolution adaptation entry in TCU.ini) and at least one TCU is already logged on:<br>The resolution is set to the resolution of the TCU which is currently active.<br>b) Dynamic resolution switching is enabled (resolution adaptation entry in tcu.ini) and no TCU has logged on yet or dynamic resolution switching is disabled:<br>The resolution is set to the max. resolution of the current PCU panel, i.e. which logged on during the previous session.<br>Default setting: Default TCU resolution in accordance with the registry.<br>**Notice:** The first TCU panel to log on is activated. The focus handler then automatically sets the resolution to this TCU panel's resolution (in the case of dynamic resolution switching).<br>B) In inactive TCU mode:<br>The resolution is not specially set - i.e. the resolution used during the previous session in the system is active, e.g. the resolution set manually in Control Panel. |
| AUTO_MON_2 | Like AUTO_MON_1, except:<br>Example 2: There is no PCU monitor, however there is a PCU panel (irrespective of whether there are any TCU panels):<br>[In active TCU mode:<br>The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]:<br>The resolution is set to the max. resolution of the PCU panel (max. 1920x1080). |
| 640x480 | During run-up, the SINUMERIK resolution is set to 640x480. |

| Settings | Meaning |
|---|---|
| 800x600 | During run-up, the SINUMERIK resolution is set to 800x600. |
| 1024x768 | During run-up, the SINUMERIK resolution is set to 1024x768. |
| 1280x1024 | During run-up, the SINUMERIK resolution is set to 1280x1024. |
| 800x480 | During run-up, the SINUMERIK resolution is set to 800x480. |
| 1280x800 | During run-up, the SINUMERIK resolution is set to 1280x800. |
| 1366x768 | During run-up, the SINUMERIK resolution is set to 1366x768. |
| 1920x1080 | During run-up, the SINUMERIK resolution is set to 1920x1080. |

## 4.11 Setting the color depth in the tcu.ini file

Per default, the color depth of the SIEMENS PC system is set to 16-bit via the `tcu.ini` configuration file and is reset to this value at every run-up.

To use a different color depth, you must adapt the setting in the `tcu.ini` file in the [VNCServer] section.

### Requirement

- Hidden directories are not shown
- A copy of the tcu.ini configuration file is stored in the `user` directory: `C:\ProgramData\Siemens\MotionControl\`**user**`\System\etc\tcu.ini`
- The `tcu.ini` configuration file in the `user` directory is not write-protected

See: Directory structure and file conventions (Page 14)

### Procedure

To adapt the color depth of the PC system, proceed as follows:

1. Open the tcu.ini configuration file in the `C:\ProgramData\Siemens\MotionControl\`**user**`\System\etc\` directory.
2. Enter the required value in the [VNCServer] section (below the `#COLOR DEPTH` comment):
   - `ColorDepth=0`
     The color depth is not changed during run-up, but rather the color depth used last is retained. In this way, you can adapt the color depth in the Windows settings.
   - `ColorDepth=1` (default value)
     The color depth is set to 16-bit during run-up.
   - `ColorDepth=2`
     The color depth is set to 32-bit during run-up.
3. Save and close the file.

## 4.12 Activating/deactivating the multitouch operation of TCUs

If you use SINUMERIK TCUs, which support capacitive multitouch operation, you can centrally activate/deactivate this functionality in the configuration file `tcu.ini` for all of the connected TCUs.

Multitouch operation is activated via a default setting.

### Requirement

- Hidden directories are not shown
- A copy of the tcu.ini configuration file is stored in the `user` directory: `C:\ProgramData\Siemens\MotionControl\`**user**`\System\etc\tcu.ini`
- The `tcu.ini` configuration file in the `user` directory is not write-protected

See: Directory structure and file conventions (Page 14)

### Procedure

Proceed as follows to configure multitouch operation:

1. Open the tcu.ini configuration file in the `C:\ProgramData\Siemens\MotionControl\`**user**`\System\etc\` directory.
2. Enter the desired value:
   - `TouchSupport=1` (default value)
     Multitouch operation is centrally activated for all of the TCUs that support the functionality.
   - `TouchSupport=0`
     Multitouch operation is centrally deactivated for all of the TCUs that support the functionality.
3. Save and close the file.

### Result

The setting has been changed and is active after a restart of the Siemens PC system.

# 4.13 Calibrating the touch operation of a screen

If you use an operator panel or screen with touch functionality, you may have to calibrate it.

The calibration of the touch functionality is always relative to the set screen resolution and is specifically saved for it after the calibration.

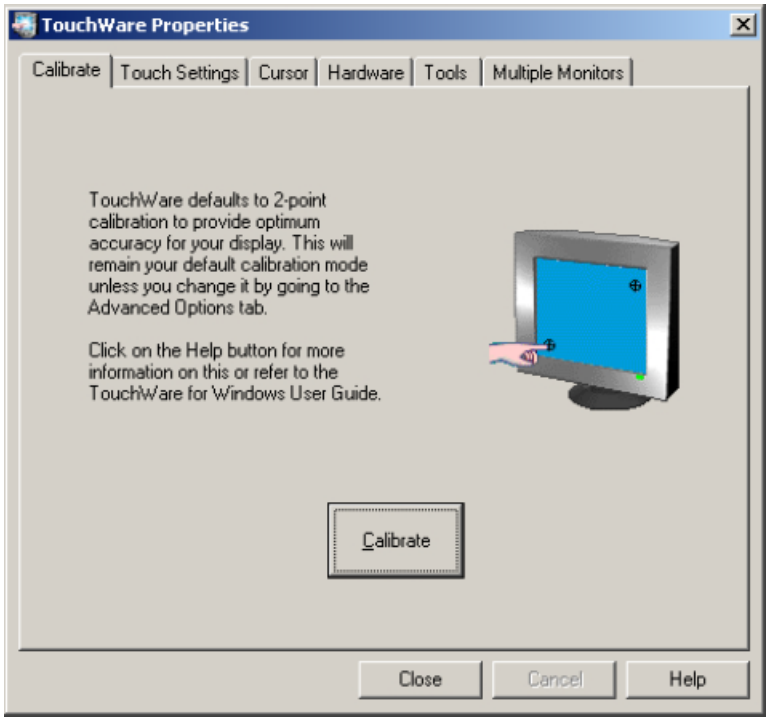
## Requirement

- The Service Desktop is active.

## Procedure

To calibrate the touch functionality of a screen, proceed as follows:

1. In the Start menu, click "Start > All Programs > Touchware > UPDD".
   The "TouchWare Properties" dialog box opens.




2. In the "Calibrate" tab, click on "Calibrate" and follow the instructions that appear on screen.
   You can find additional information in the dialog via the "Help" button.

# 4.14 Configuring virtual direct control keys (VDCK)

## 4.14.1 General information

Direct control keys allow key states to be directly transferred to the PLC. These key signals are transferred in a 2-byte key image and are evaluated by the PLC as digital inputs. Depending on the panel being used, there are different options for using direct control keys via the system network, e.g. using a direct control key module or machine control panel.

### Virtual direct control keys (VDCK)

The PCU Base Software extends the direct key configuration options to include virtual direct control keys (**V**irtual **D**irect **C**ontrol **K**eys).

VDCK supports multitouch operation and therefore supplements the new SINUMERIK and SIMATIC operating concept to include the corresponding virtual direct control keys.

**Functional scope**

| | |
|---|---|
| Maximum number of supported direct control keys: | 16 |
| Direct control keys that can be simultaneously actuated: | Depending on the multitouch functionality of the panel being used, e.g.:<br>• SINUMERIK OP 015 black: 10 touch points<br>• SINUMERIK OP 019 black: 5 touch points<br>• SINUMERIK TOP 1500/1900/2200: 5 touch points<br>• SIMATIC ITC 1500/1900/2200 V3: 5 touch points<br>• SIMATIC IFP 1500/1900/2200: 5 touch points<br>• SIMATIC IPC 277/477/677: 5 touch points<br>You can find an overview of the compatible SIMATIC configurations in the section titled "PCU Base Software Windows 7/10 (Page 7)". |
| Layout of the direct control keys | 1 global direct control key layout for all HMI operating screens |
| Assignment of the direct control keys | Direct control keys 1…16 are permanently assigned bits (0…15) in the image. Deactivating individual direct control keys (or using less than 16 direct control keys) does not influence the assignment to the relevant bits. |
| Signal transfer | 1-channel |

| NOTICE |
|---|
| **Signals of the virtual direct control keys are transferred from the HMI to the PLC through 1 channel**<br>In addition, use a permanently wired acknowledgment button or a 2nd Channel from the HMI (HMI Pro only) in order to guarantee functional safety. |

**PLC basic program functionality**

The direct control key functionality in the PLC basic program (activating the data transfer, interface signal, addressing, alarms etc.) is identical when using virtual direct control keys and real direct control keys.

You can find additional information in the SINUMERIK 840D sl Function Manual Basic Functions (FB1), Chapter "Configuring the machine control panel, handheld terminal, direct control keys".

## 4.14.2 Creating a configuration file

You can define the layout of the virtual direct control keys in the associated configuration file. To do this, you must create the configuration file in the user directory on the system which is the active boot server.

### Requirement

- The direct control keys are activated in the operator panel settings.
- A valid TCU index is configured in the operator panel settings.

You can find additional information about these settings in the corresponding manual.

### Procedure

Proceed as follows to create the configuration file for the direct control key layout:

1. Open a text editor, and as character coding, set ASCII or ANSI.
2. Copy the content from Reference to the vdck.ini configuration file (Page 48) to the text editor.
3. Save the file as `vdck.ini` in one of the user directories of the system that is the active boot server.
   PC system as active boot server:
   - `C:\ProgramData\Siemens\MotionControl\user\common\tcu\<Name of the TCU>\common\tcu\vdck.ini`
   - `C:\ProgramData\Siemens\MotionControl\oem\common\tcu\<Name of the TCU>\common\tcu\vdck.ini`
   - `C:\ProgramData\Siemens\MotionControl\addon\common\tcu\<Name of the TCU>\common\tcu\vdck.ini`

   NCU as active boot server:
   - `/user/common/tcu/`<Name of the TCU>`/common/tcu/vdck.ini`
   - `/oem/common/tcu/`<Name of the TCU>`/common/tcu/vdck.ini`
   - `/addon/common/tcu/`<Name of the TCU>`/common/tcu/vdck.ini`

   See also: Directory structure and file conventions (Page 14)

### Result

The configuration file has been created in a user directory and can be adapted.

You will find further information on the setting options in the following chapters.

## 4.14.3 Adapting the layout of the virtual direct control keys

You can adapt the following settings of the virtual direct control key layout.

Please note that the direct control key numbering (Page 48) and the assignment to the appropriate bits in the image is fixed and cannot be changed. Moving or deactivating a control key does not change the assignment or numbering.

### Precondition

- Configuration file vdck.ini is saved in one of the user directories, and opened in an editor. See: Creating a configuration file (Page 46), Reference to the vdck.ini configuration file (Page 48)

### Adapting the position of a direct control key

1. In vdck.ini adapt the required value for the horizontal (x) and vertical (y) positions, and note the following:
   - The control key positions are entered in the vdck.ini in the corresponding sections under X_POS_<control key number> and Y_POS_<control key number>.
   - The positioning of the individual control keys is specified in per mill and not in pixels. This means that the direct control key layout is independent of the screen resolution.
   - When parameterizing, orientate yourself using the diagram (Page 48).
2. If necessary, make other modifications or save and close the file.

### Adapting the size of a direct control key

1. In vdck.ini adapt the required height and width, and note the following:
   - In vdck.ini, width and height are entered in the corresponding sections under WIDTH_<control key number> and HEIGHT_<control key number>.
   - The size of the individual control keys is specified in per mill and not in pixels. This means that the direct control key layout is independent of the screen resolution.
   - In the default setting, every control key has a size of 146 x 112 per mill (width x height).
   - When parameterizing, orientate yourself using the diagram (Page 48).
2. If necessary, make other modifications or save and close the file.

### Hiding a direct control key

1. For the corresponding direct control key, change the values for height and width to 0. For example, to hide direct control key 1, set the following values:
   ```
   WIDTH_1 = 0
   HEIGHT_1 = 0
   ```
2. If necessary, make other modifications or save and close the file.

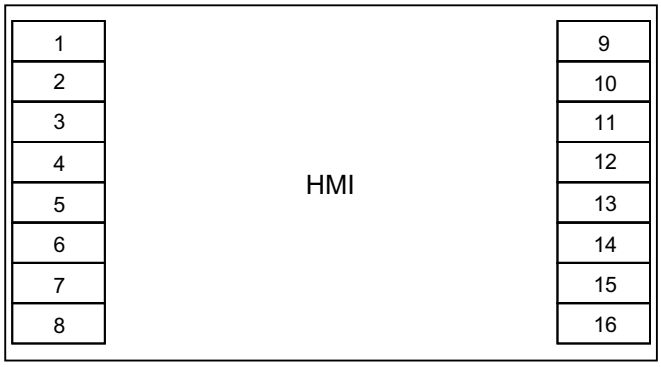
## 4.14.4 Default setting for the direct control key layout

In the configuration file vdck.ini, the direct control keys are defined and positioned as follows in the default setting.

**Note**

**Use of the virtual direct control keys by HMI Pro**

Please note that HMI Pro does not use the 4 outer virtual direct control keys (direct control keys 1, 8, 9 and 16).




## 4.14.5 Reference to the vdck.ini configuration file

Configuration file vdck.ini contains the settings of the virtual direct control key layout.

The individual virtual direct control keys are defined in per mill and not in pixel. This means that you can use the same direct control key layout for several operator panels with different resolutions.

**Archive**

The file must be located on the active boot server in the directory of the TCU.

Example for a PCU 50.5: C:\ProgramData\siemens\MotionControl\user\common\tcu\<Name of the TCU>\common\tcu\vdck.ini

Example for an NCU: /user/common/tcu/<Name of the TCU>/common/tcu/vdck.ini

See also: Directory structure and file conventions (Page 14)

### Character coding to be used

ASCII

### Configuration file for the default layout

**vdck.ini**

```
[VDCK]
NUM_LAYOUTS = 1

[LAYOUT_1]
# Number of keys
NUM_KEYS = 16
# X positions in per mill of the resolution in the width
X_POS_1 = 0
X_POS_2 = 0
X_POS_3 = 0
X_POS_4 = 0
X_POS_5 = 0
X_POS_6 = 0
X_POS_7 = 0
X_POS_8 = 0
X_POS_9 = 854
X_POS_10 = 854
X_POS_11 = 854
X_POS_12 = 854
X_POS_13 = 854
X_POS_14 = 854
X_POS_15 = 854
X_POS_16 = 854
# Y positions in per mill of the resolution in the height
Y_POS_1 = 49
Y_POS_2 = 161
Y_POS_3 = 273
Y_POS_4 = 385
Y_POS_5 = 499
Y_POS_6 = 611
Y_POS_7 = 724
Y_POS_8 = 836
Y_POS_9  = 49
Y_POS_10 = 161
Y_POS_11 = 273
Y_POS_12 = 385
Y_POS_13 = 499
Y_POS_14 = 611
Y_POS_15 = 724
Y_POS_16 = 836
```

**vdck.ini**

```
# Width per mill of the resolution
WIDTH_1 = 146
WIDTH_2 = 146
WIDTH_3 = 146
WIDTH_4 = 146
WIDTH_5 = 146
WIDTH_6 = 146
WIDTH_7 = 146
WIDTH_8 = 146
WIDTH_9 = 146
WIDTH_10 = 146
WIDTH_11 = 146
WIDTH_12 = 146
WIDTH_13 = 146
WIDTH_14 = 146
WIDTH_15 = 146
WIDTH_16 = 146
# Height in per mill of the resolution
HEIGHT_1 = 112
HEIGHT_2 = 112
HEIGHT_3 = 112
HEIGHT_4 = 112
HEIGHT_5 = 112
HEIGHT_6 = 112
HEIGHT_7 = 112
HEIGHT_8 = 112
HEIGHT_9 = 112
HEIGHT_10 = 112
HEIGHT_11 = 112
HEIGHT_12 = 112
HEIGHT_13 = 112
HEIGHT_14 = 112
HEIGHT_15 = 112
HEIGHT_16 = 112
```

# 4.15 Configuring the SITOP UPS module for use with the PCU Base Software

## 4.15.1 Overview of SITOP UPS

SITOP UPS modules can continue operation of the SIEMENS PC system temporarily if a power failure occurs and/or shut down the system correctly.

For example, a SITOP UPS module also protects against data loss when the SIEMENS PC system hardware is shut down, because for technical reasons with Windows-based systems, data is still written to the SSD after shutdown.

- You will find general information about SITOP UPS modules used in combination with the PCU base software in Chapter SITOP modules for the PCU 50.5 (Page 52).
- If you want to use a SITOP UPS module in combination with the PCU base software, you must first adapt the settings of the SITOP software (Page 53).
- Optionally, you can adapt the delay time after which the HMI software will be forced to shut down (Page 56).
- You must configure the hardware (Page 59) of the SITOP UPS module before using it with the PCU base software.

| NOTICE |
| --- |
| **Ensure the readiness for operation of the SITOP power supply**<br>In order to prevent data loss, the SITOP power supply must be ready for operation. Note the following:<br>• The SITOP power supply may not be ready for operation immediately after being switched on (a capacitor-buffered UPS500 requires 1-2 minutes make time, for example).<br>• The operating system and the SITOP software must have run up so that the SITOP software can shut down the system correctly when a power failure occurs.<br>• The readiness for operation and the functional capability of the SITOP power supply must be ensured (e.g. battery of capacitor charged).<br>• Please also note all the information on the readiness for operation in the documentation of your SITOP device. |

### Further information

Information on the use of SITOP UPS modules in combination with the PCU base software can also be found on the Internet:

- Service & Support Portal: SINUMERIK PCU 50.5: Use of an uninterruptible power supply (http://support.automation.siemens.com/WW/view/en/90142681)
- Service & Support Portal: Typical values for an operator station with PCU 50.5 Windows 7 (http://support.automation.siemens.com/WW/view/en/76773241)
- SITOP Selection Tool (http://www.siemens.com/sitop-selection-tool) (Tool for the selection of a SITOP UPS module)

## 4.15.2 SITOP modules for the PCU 50.5

### Suitable SITOP UPS modules

SITOP UPS modules can continue operation of a PC system temporarily if a power failure occurs and/or shut down the system correctly.

In conjunction with the PCU base software, all specific applications are taken into account (e.g. HMI software) with the `USVShutdown.bat` component.

Safe shutdown after a power failure of the PCU 50.5 as delivered takes approximately 20 seconds. For a typical power consumption of approx. 60 W, a capacitor-buffered UPS with an energy storage device of 2.5 kW is recommended for a PCU 50.5. If longer buffer times are required, then several expansion modules can be connected in a cascade connection.

**Example of a SITOP UPS module that can be used (USB):**

SITOP 500S (15 A / 2.5 kWs capacitor)

Article no.: 6EP1933-2EC41

**Example of a SITOP UPS with Ethernet/PROFINET interface that can be used:**

SITOP UPS1600 starter kit

- DC-UPS SITOP UPS1600 24V DC/10A
- SITOP UPS1100 3.2 AH battery module

Article No.: 6EP4134-3AB00-2AP0

---

**Note**

**Information about the test environment for machine OEMs**

The functionality has been tested in the standard configuration with the PCU base software.

When installing add-on or OEM software components, you must check the shutdown procedure of the entire system and adapt the factory setting (180 seconds).

See: Parameterizing a delay time for quitting the HMI software (Page 56)

---

### Further information

Information about available SITOP UPS modules is given in the catalog or in the Internet:

- Service & Support Portal: SINUMERIK PCU 50.5: Use of an uninterruptible power supply (http://support.automation.siemens.com/WW/view/en/90142681)
- Service & Support Portal: Typical values for an operator station with PCU 50.5 Windows 7 (http://support.automation.siemens.com/WW/view/en/76773241)
- SITOP Selection Tool (http://www.siemens.com/sitop-selection-tool) (Tool for the selection of a SITOP UPS module)
- Catalog KT 10.1 Power supply SITOP
- SIEMENS Industry Mall > 24 V DC Uninterruptible Power Supplies (https://eb.automation.siemens.com/mall/en/US/Catalog/Products/7010117)

## 4.15.3 Configuring SITOP software for PCU Base Software

### 4.15.3.1 Configuring SITOP software V3.x (USB)

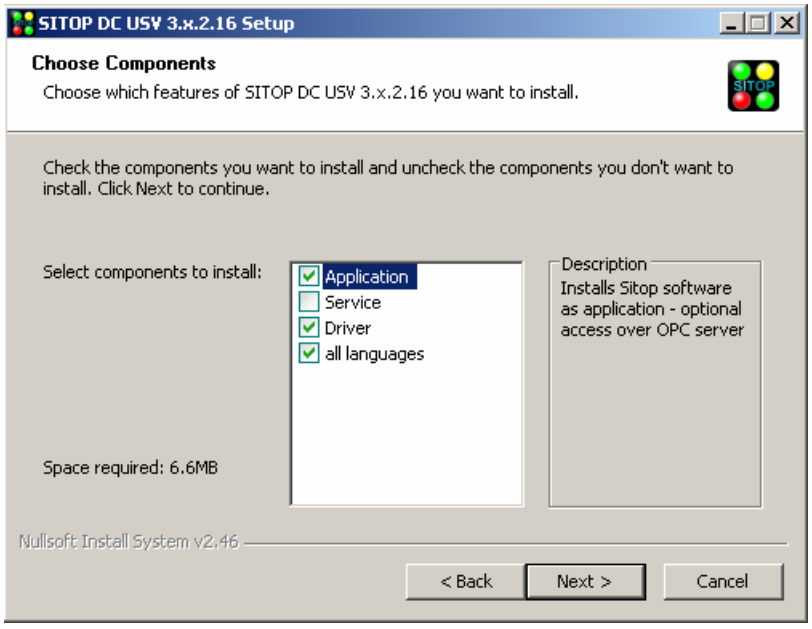
#### Configuring SITOP software V3.x (USB)

If you want to use a SITOP UPS module in combination with the PCU base software, you must first adapt the settings of the SITOP software.

#### Requirement

- The SITOP UPS hardware is connected via the USB interface.
- The SITOP software of version 3.x.2.16 or later is installed in directory `C:\Program Files (x86)\SITOP\`

- If you want to use SITOP software of version 3.2.1.16 or older, this must be installed as a normal application, not as a service.




**Note**

**Windows service of the SITOP software version 3.2.1.16 or older is not compatible with the PCU base software**

If you use SITOP software of version 3.2.1.16 or older as a Windows service with the PCU base software, the correct shutdown procedure cannot be performed!

If you have installed the SITOP software of version 3.2.1.16 as a Windows service, you must uninstall this and install it again as an application in order to be able to use it in conjunction with the PCU base software.

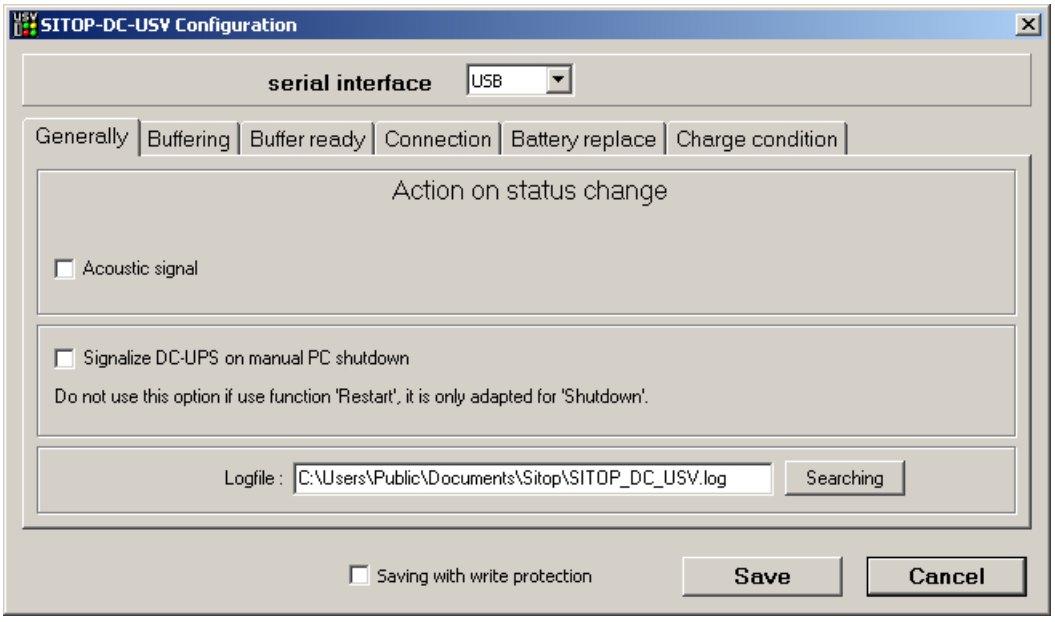
As of version 3.2.1.17, the SITOP software can be used as service with the PCU base software and is recommended.

- The SITOP software was configured during the installation so that it starts automatically when Windows is run-up (factory setting).
- The Service Desktop is active.

## Procedure

Proceed as follows to configure the SITOP software for use with the PCU base software:

1. Call the settings of the SITOP software, for example, by right-clicking the SITOP UPS icon in the Windows information area and then selecting "Configuration".
   The "SITOP DC UPS Configuration" dialog box opens.




2. In the "General" tab, select "UPS" in the drop-down list at "Serial interface".

3. Switch to the "Floating operation" tab and make the following settings:
   - Deactivate the "Display monitoring window after" checkbox

     **Note**
     **Deactivating the monitoring window**

     Display of the monitoring window can cause malfunctioning of the HMI software.

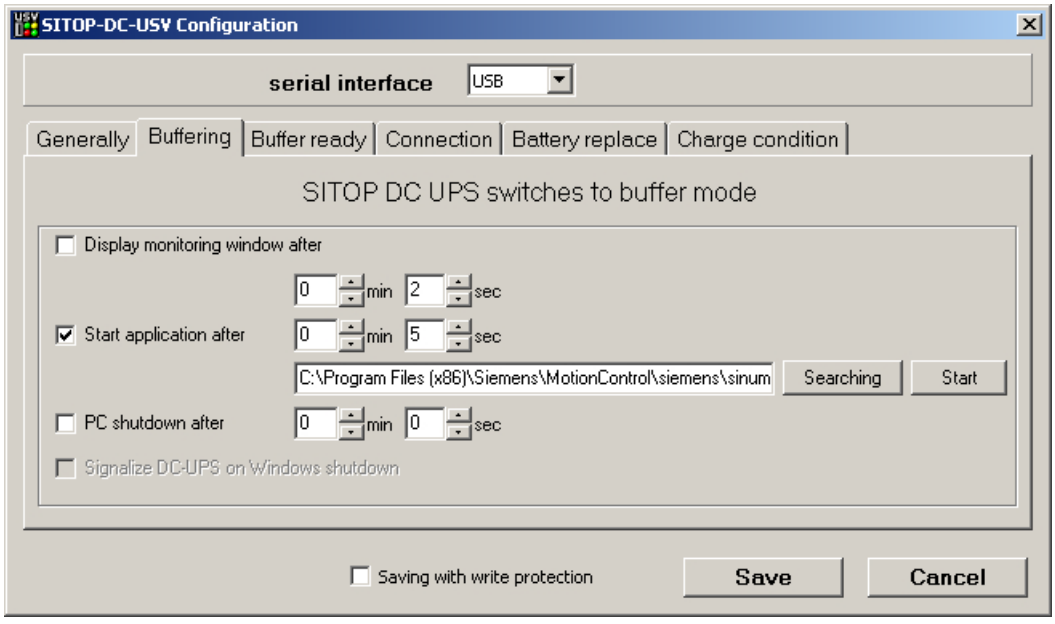
   - Click directly in the text field at "Start application after" and specify the path of the `USVShutdown.bat`:
     `C:\Program Files (x86)\Siemens\MotionControl\siemens\sinumerik\hmi\base\USVShutdown.bat`

     **Note**
     **Do not use the "Browse" button**

     The dialog can crash if you use the "Browse" button to specify the path of the `USVShutdown.bat`.

     Instead, enter the path directly in the text field.

   - After you have specified the path of the `USVShutdown.bat`, activate the "Start application after" checkbox. Optionally, you can specify the wait time after which the `USVShutdown.bat` is to be executed (e.g. 5 seconds).




4. Confirm the settings with "Save".

## Parameterizing a delay time for quitting the HMI software

Before the SITOP monitor closes Windows 7, `USVShutdown.bat` shuts down operation of the HMI software.

If the HMI software cannot be shut down within 180 seconds (factory setting) due to an error, HMI software shutdown is forced and Windows 7 is shut down.

If the HMI software of your OEM installation is not shut down within 180 seconds, you can parameterize this wait time manually.

**Note**

**Information about the test environment for machine OEMs**

The "SITOP UPS" function has been tested in the standard configuration with the PCU base software. When installing add-on or OEM software components, you must check the shutdown procedure of the entire system and adapt the factory setting.

## Requirement

- The SITOP UPS module is configured for use with the PCU base software.
  See Configuring SITOP software V3.x (USB) (Page 53).
- The Service Desktop is active.

### Procedure

Proceed as follows to change the delay time:

1. Call the settings of the SITOP software, for example, by right-clicking the SITOP UPS icon in the information area and then selecting "Configuration".
2. In the "SITOP DC UPS Configuration" dialog box, enter the wait time in seconds as a command line parameter in the "Floating operation" tab:
   - Syntax: `<Path>\USVShutdown.bat -<Time in seconds>`
   - Example: `C:\Program Files (x86)\Siemens\MotionControl\siemens\sinumerik\hmi\base\USVShutdown.bat -180`

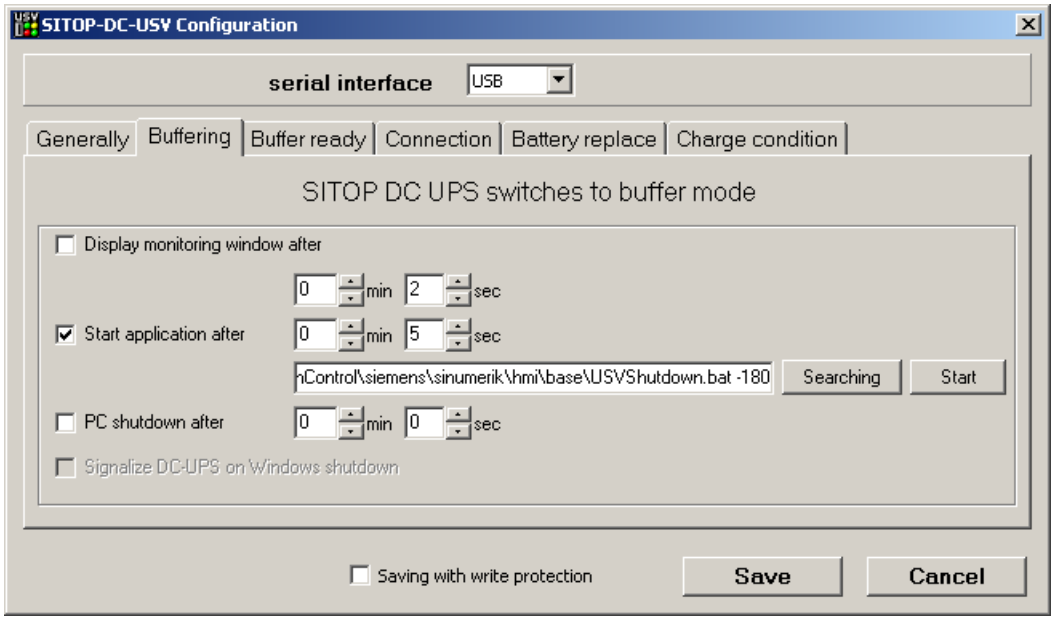



Figure 4-1 SITOP Monitor Floating operation

3. Confirm the setting with "Save".

## 4.15.3.2 Configuring SITOP UPS Manager (Ethernet)

### Precondition

- The SITOP UPS hardware is connected via the Ethernet interface.
- The SITOP-UPS Manager has been installed and configured for Ethernet.

**Note**

**Validity of the information on the SITOP components in this manual**

This information only involves the configuration of the settings relevant for the PCU base software. In addition, all the information and notes in the corresponding manuals of your SITOP UPS hardware and SITOP UPS Manager apply.

### Procedure

1. Call the SITOP UPS Manager.
2. Change to "Software configuration > Buffer mode".
3. Make the following settings:
   - Activate the checkbox "Start application on power failure" - and there, enter the path of USVShutdown.bat:
     C:\Program Files (x86)\Siemens\MotionControl\siemens\sinumerik\hmi\base\USVShutdown.bat
   - When required, configure the wait time until the HMI software is terminated by USVShutdown.bat (factory setting = 180 seconds).

     **Note**

     **Information about the test environment for machine OEMs**

     The "SITOP UPS" function has been tested in the default configuration with the PCU base software. When installing add-on or OEM software components, you must check the shutdown procedure of the entire system and adapt the factory setting.

     To do this, enter the wait time in seconds as command line parameter:
     Syntax: <Pfad>\USVShutdown.bat -<Time in seconds>
     Example: C:\Program Files (x86)\Siemens\MotionControl\siemens\sinumerik\hmi\base\USVShutdown.bat -180
4. Confirm the settings with "Configuration changed, restart application now".

## 4.15.4 Hardware configuration of the SITOP UPS module

### Buffering parameterization

The UPS module can be used to select whether buffering should be completed after a predetermined period of time or not until the accumulator's lowest discharge threshold (= maximum buffer time) has been reached. Both buffering parameterizations result from this.

### "Maximum buffer time" mode

This mode enables the system to be shut down in a time-optimized manner. The UPS module is synchronized with the shutdown of the operating system. Buffering is maintained until the operating system has been shut down. The operating system must shut down within a maximum of five minutes (including all applications). Otherwise, the UPS module buffers for the maximum buffer time (dependent on the accumulator state).

**Required settings on the UPS module (USB interface)**

| | On - Off | | | |
|---|---|---|---|---|
| 1 | | ○ | +2V | Cut-in threshold<br>+22V fixed |
| 2 | | ○ | +1V | |
| 3 | ○ | | +0.5V | |

| | On - Off | | | |
|---|---|---|---|---|
| 4 | | ○ | +1V | End-of-charge voltage<br>+ 26.3V fixed |
| 5 | | ○ | +1V | |
| 6 | ○ | | +0.5V | |
| 7 | ○ | | +0.2V | |
| 8 | | ○ | +0.2V | |
| 9 | | ○ | +0.1V | |
| 10 | | ○ | 0.35A / 0.7A | Charging current |

| | On - Off | | | |
|---|---|---|---|---|
| 1 | | ● | | Set time/max. time |
| 2 | | ○ | +320 s | Buffer time<br><br>+5 s fixed |
| 3 | | ○ | +160s | |
| 4 | | ○ | +80s | |
| 5 | | ○ | +40s | |
| 6 | | ○ | +20s | |
| 7 | | ○ | +10s | |
| 8 | ● | | | Disconnection |
| 9 | | ○ | | Battery operating state on/off |

Legend:

○ Delivery condition setting

● Setting for operation on the PCU

## "Fixed buffer time" mode

In this mode, the UPS module always buffers for the pre-selected, fixed period of time. It is not possible to synchronize the UPS module with the operating system shutdown.

### Required settings on the UPS module

| | On - Off | | | |
|---|---|---|---|---|
| 1 | | ○ | +2V | Cut-in threshold<br>+22V fixed |
| 2 | | ○ | +1V | |
| 3 | ○ | | +0.5V | |

| | On - Off | | | |
|---|---|---|---|---|
| 4 | | ○ | +1V | ◦<br>◦<br>End-of-charge voltage<br>+ 26.3V fixed |
| 5 | | ○ | +1V | |
| 6 | ○ | | +0.5V | |
| 7 | ○ | | +0.2V | |
| 8 | | ○ | +0.2V | |
| 9 | | ○ | +0.1V | |
| 10 | | ○ | 0.35A / 0.7A | Charging current |

| | On - Off | | | |
|---|---|---|---|---|
| 1 | ● | | | Set time/max. time |
| 2 | | ○ | +320 s | ◦<br>◦<br>Buffer time<br>+5 s fixed |
| 3 | ● | | +160s | |
| 4 | | ○ | +80s | |
| 5 | | ○ | +40s | |
| 6 | | ○ | +20s | |
| 7 | | ○ | +10s | |
| 8 | ● | | | Disconnection |
| 9 | | ○ | | Battery operating state on/off |

Legend:

○ Delivery condition setting

● Setting for operation on the PCU

# 4.16 Configuration of the Service Center

## 4.16.1 Overview

You can permanently adapt the settings for the backup and restoration of disk images (Service Center (Page 121) component) in the "Service Center Backup/Restore" dialog box:

- Configuration of the network adapter (Page 63)
- Configuration of the host (Page 64)

Adaptations that you make here are stored in the `servicesystem.ini` configuration file.

You can also adapt the settings directly in the associated configuration files:

- `servicesystem.ini`
- Additional settings: `ghost.ini`

You can also view the log files of the Service Center or restart the PC system in the Service Center in the "Service Center Backup/Restore" dialog box.

To open the dialog box, click the ServiceCenter Backup-Restore icon on the Desktop.

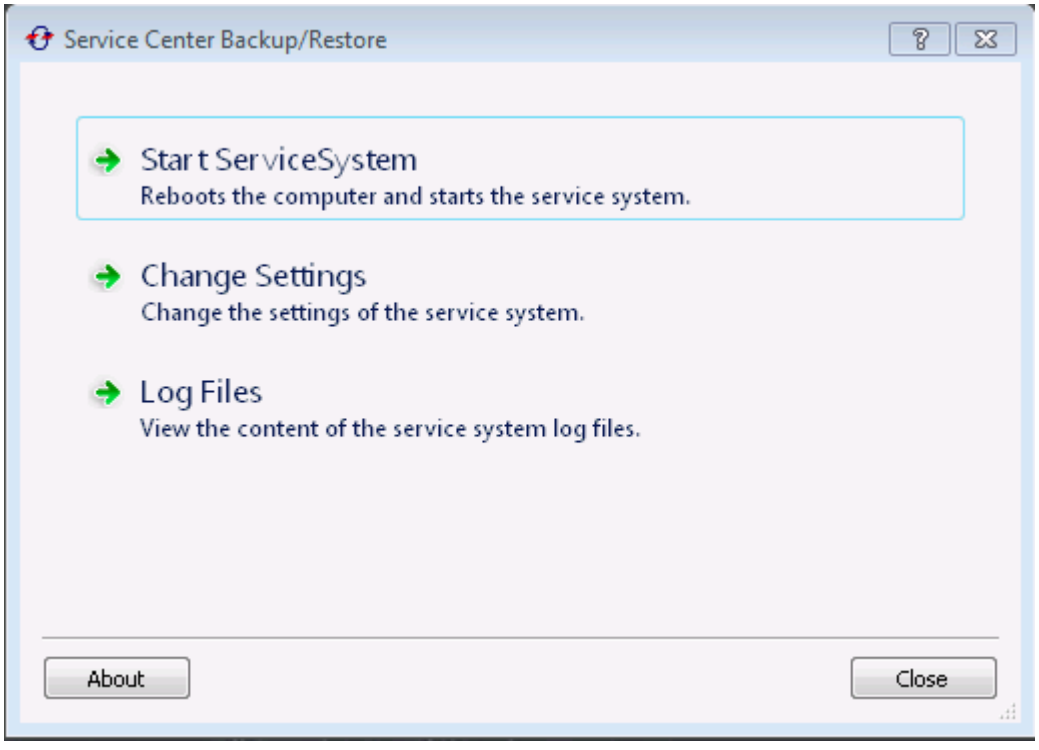



Figure 4-2 ServiceCenter Backup/Restore

## 4.16.2 Configuration of the network adapter

You can make the network settings in the "Adapter Settings" tab and permanently save this configuration, in contrast to the settings in the "Network Settings" dialog box of the Service Center (Page 125).

Settings that you make in this dialog box are stored in the `servicesystem.ini` configuration file.

### Overview

You can make the following IP settings in the "Adapter Settings" dialog box:

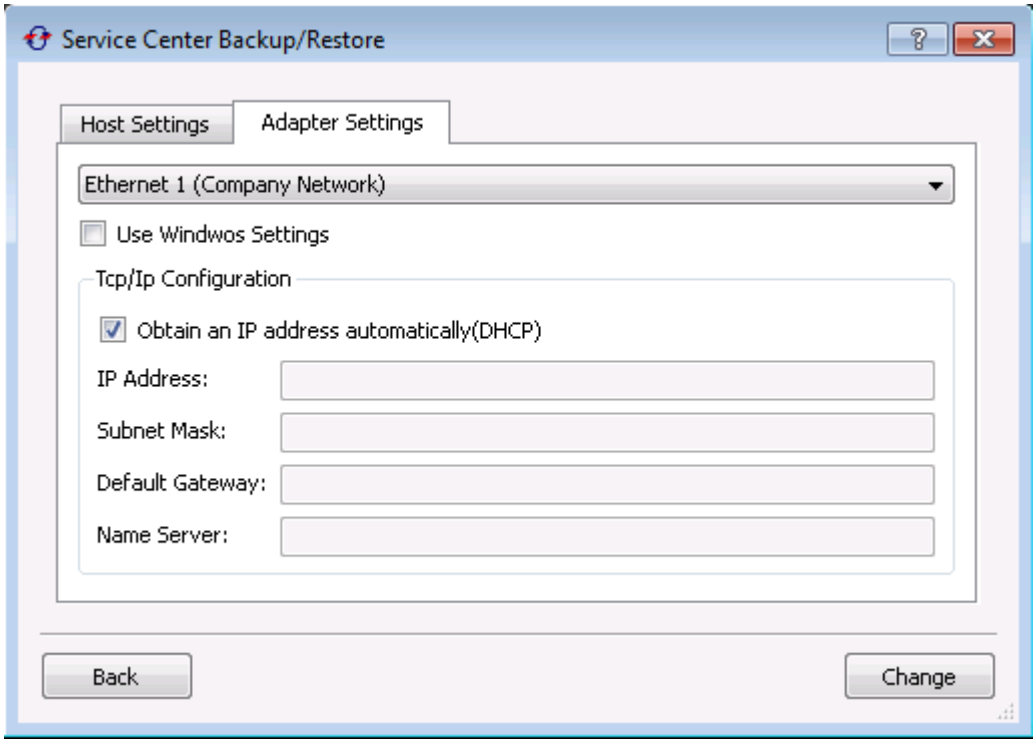



Figure 4-3 Adapter settings - Ethernet 1 (company network)

Table 4-4 Settings in the "Adapter Settings" dialog box

| Settings | Purpose |
|---|---|
| Adapter | Select the Ethernet interface that you want to configure.<br>All settings in this dialog box refer to the selected Ethernet interface. |
| Use Windows Settings | Select whether the values set in the Control Panel are to be used. |
| Obtain an IP address automatically (DHCP) | Select whether the addresses are to be fetched dynamically from the DHCP server.<br>If the checkbox is deactivated, you must make the settings manually. |
| IP Address | Set the IP address of the NCU.<br>You can specify an IP address from the following range:<br>192.168.214.250 - 254 |
| Subnet Mask | Specify a subnet mask, e.g. 255.255.255.0 |

| Settings | Purpose |
|---|---|
| Default Gateway | Set the IP address of the standard gateway.<br>A standard gateway creates a standard route in the IP routing table for all destinations that are not in the subnet. |
| Name Server | Set the IP address of the name server.<br>A name server answers questions asked about a domain name zone using a DNS database. |

## 4.16.3 Configuration of the host

You can make the network settings in the "Host Settings" tab and permanently save this configuration, in contrast to the settings in the "Network Settings" dialog box of the Service Center.

Settings that you make in this dialog box are stored in the `servicesystem.ini` configuration file.

### Overview

You can make the following settings in the "Host Settings" dialog box:

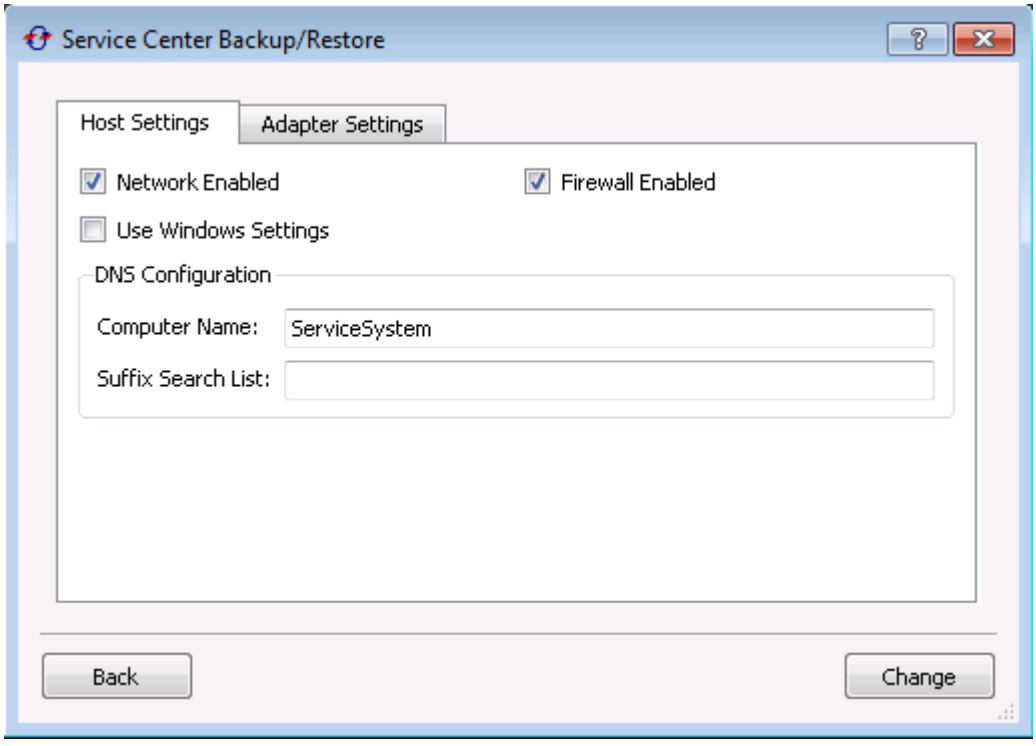



Figure 4-4 Host settings

Table 4-5 IP settings in the "Host Settings" dialog box

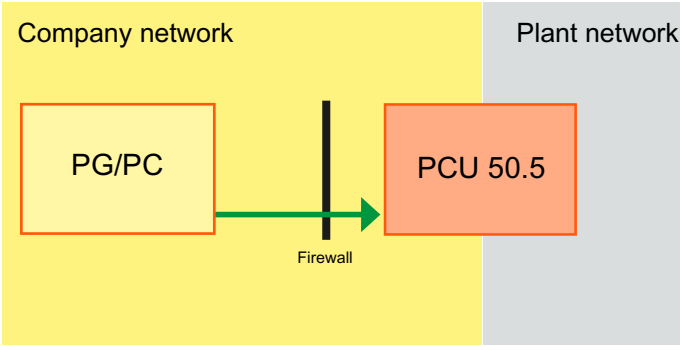
| Settings | Purpose |
|---|---|
| Network enabled | Activate or deactivate the network interfaces of the PC system. |
| Firewall enabled | Activate or deactivate the firewall of the PC system. |
| Use Windows Settings | Select whether the values set in the Control Panel are to be used. |
| DNS Configuration > Computer Name | Specify the DNS computer name. |
| DNS Configuration > Suffix Search List | Parameterize a search list for DNS suffix, e.g. "network.com".<br>The factory setting of the Ethernet interface "Local Area Connection" is ".local". |

# 4.17 Adapting the firewall settings

## 4.17.1 Overview

From a PG/PC in the plant network, you can directly access resources on the Siemens PC system. The same applies when accessing a PG/PC from the Siemens PC system in the plant network or in the company network.

If you want to access the Siemens PC system via the company network instead or in addition, you must adapt the firewall settings.




You can adapt the following settings:

- Activate the remote access to the PC system (Page 72)
- Enabling SNMP communication (Page 70)
- Activation/deactivation of the file and printer release (Page 75)
- Activation of the ping execution (ICMP) (Page 87)

As a rule, you can make these settings in the Control Panel or via the prompt or a script/batch file (relevant for series commissioning) (Page 69).

You can save and restore (Page 89) the firewall settings. (Even on different Siemens PC systems, e.g. due to series commissioning)

Also note the general information in this chapter:

- Factory setting of the Windows Firewall on the Siemens PC system (Page 67)
- General information on the operating principle of the Windows Firewall (Page 68)
- General information on the settings recommended for Windows (Page 69)

## 4.17.2 Factory setting of the Windows Firewall on the Siemens PC system

The Windows Firewall settings on your Siemens PC system are appropriately configured by the PCU-Basesoftware for the use of SINUMERIK software and systems. For security reasons, the Windows Firewall is activated in the company network and expanded by additional rules. The Windows Firewall is deactivated in the system network. The descriptions in this chapter therefore largely relate to the adaptation of the firewall settings for the company network.

Table 4-6 Factory settings for network adapter and Windows Firewall

| | System network | Company network |
|---|---|---|
| Interface | X2 | X1 |
| Network name for PCU 50.5 | Local Area Connection | Local Area Connection 2 |
| Network profile | Public | Public |
| Status Windows Firewall<br>See: General information on the settings recommended for Windows (Page 69) | Deactivated | Activated |
| File and printer release<br>See: Activation/deactivation of the file and printer release (Page 75) | Activated and permitted | Deactivated and blocked by firewall |
| SNMP communication<br>See: Enabling SNMP communication (Page 70) | Permitted | Blocked |
| Remote access (via port 5900)<br>See: Activate the remote access to the PC system (Page 72) | Permitted | Blocked |
| Outgoing ICMP Ping requests from the PC system<br>See: Activation of the ping execution (ICMP) (Page 87) | Permitted | Permitted |
| Incoming ICMP Ping requests on the PC system<br>See: Activation of the ping execution (ICMP) (Page 87) | Permitted | Blocked |

**Note**

**Automatic network release of USB data storage media results in activation of the file and printer release**

The automatic network release of USB data storage media is activated in the default setting of the NCU-Basesoftware. The file and printer release is then activated by Windows when a USB data storage medium is connected.

See: Configuring network access on the USB data storage medium (Page 34)

## See also

Patch management and security updates (Page 96)

Saving and restoring firewall settings (Page 89)

### 4.17.3 General information on the operating principle of the Windows Firewall

Windows Firewall rules are structured as follows:

- The firewall rules are divided into Inbound Rules and Outbound Rules.
- Each firewall rule is valid for one or several network profiles (Profiles).
  Each network connection in Windows is assigned to a specific network profile. In the factory settings state of the PC system, both the plant network ("Local Area Connection") and the company network ("Local Area Connection 2") have the network profile "Public".
- Some firewall rules are already assigned to so-called "Groups". As a result, you can more easily activate or deactivate specific related network functionalities (e.g. file and printer release).
- To ensure that the compatibility with Windows XP systems exists in the network, the Windows XP Firewall settings still exist in Windows 7 in the "Local Group Policy Editor" under "Computer Configuration > Administrative Templates > Network > Network Connections > Windows Firewall".
  If you make changes at this point, the settings take effect in addition to the settings of the Windows 7 Firewall. Conflicts can occur if the configuration is erroneous. Do not make any changes to the Windows XP settings if there are no Windows XP systems in your network.

You can make the Windows Firewall settings both via the Control Panel and via the prompt or script/batch files. The use of script/batch files is particularly recommended for series commissioning. Alternatively, for series commissioning, you can make all of the Windows Firewall settings on an individual PC system, save these settings, and apply them on other devices (Page 89).

## 4.17.4 General information on the settings recommended for Windows

### Firewall settings recommended by Windows are unsuitable for use in the plant network

In the Control Panel, under "Windows Firewall", there is a display indicating that the firewall settings do not correspond to the settings recommended and automatically checked by Windows. The corresponding sections are highlighted in red:




The settings automatically recommended by Windows are not suitable for use in the plant network, however, and would limit the functioning of the system. The network security in the plant network must therefore be ensured by other security precautions. In this regard, observe the information on industrial security (Page 96).

**Note**

**Do not use the firewall settings that are automatically recommended by Windows!**

The command "Use recommended settings" overwrites the existing firewall settings, which are required for operating the PCU-Basesoftware, and therefore this command must not be used.

Otherwise, the Windows Firewall will be activated in the plant network and communication of internal services of the PCU-Basesoftware will be inhibited.

## 4.17.5 Configuration by means of a prompt or script/batch file

You can make most of the settings described in this chapter both via the Control Panel and via the prompt or script/batch files. The use of script/batch files is particularly recommended for series commissioning.

### Basic procedure

You can enter these commands either directly into the prompt or save the command in a script or batch file:

- **Direct input of the command:**
  - Call the prompt as the administrator.
  - You can enter the commands either directly or in succession.
- **Calling the command via a script file (or batch file):**
  - Save the command in a text file.
  - Call the prompt as the administrator.
  - Specify the script file as follows:
    `netsh -f <File name of the script file>`

### Further information

- MSDN Library - Netsh (command-line program) (https://msdn.microsoft.com/en-us/library/cc785383(v=ws.7).aspx)

## 4.17.6 Enabling SNMP communication

Some network diagnostics functions use SNMP communication (e.g. in SINUMERIK Operate). This is not blocked in the plant network. If, however, you have productively linked your plant in the company network, SNMP communication is partially blocked:

- In the factory settings version of the PCU-Basesoftware, incoming SNMP communication is blocked in the company network.
- Outgoing SNMP communication in the company network is not blocked.

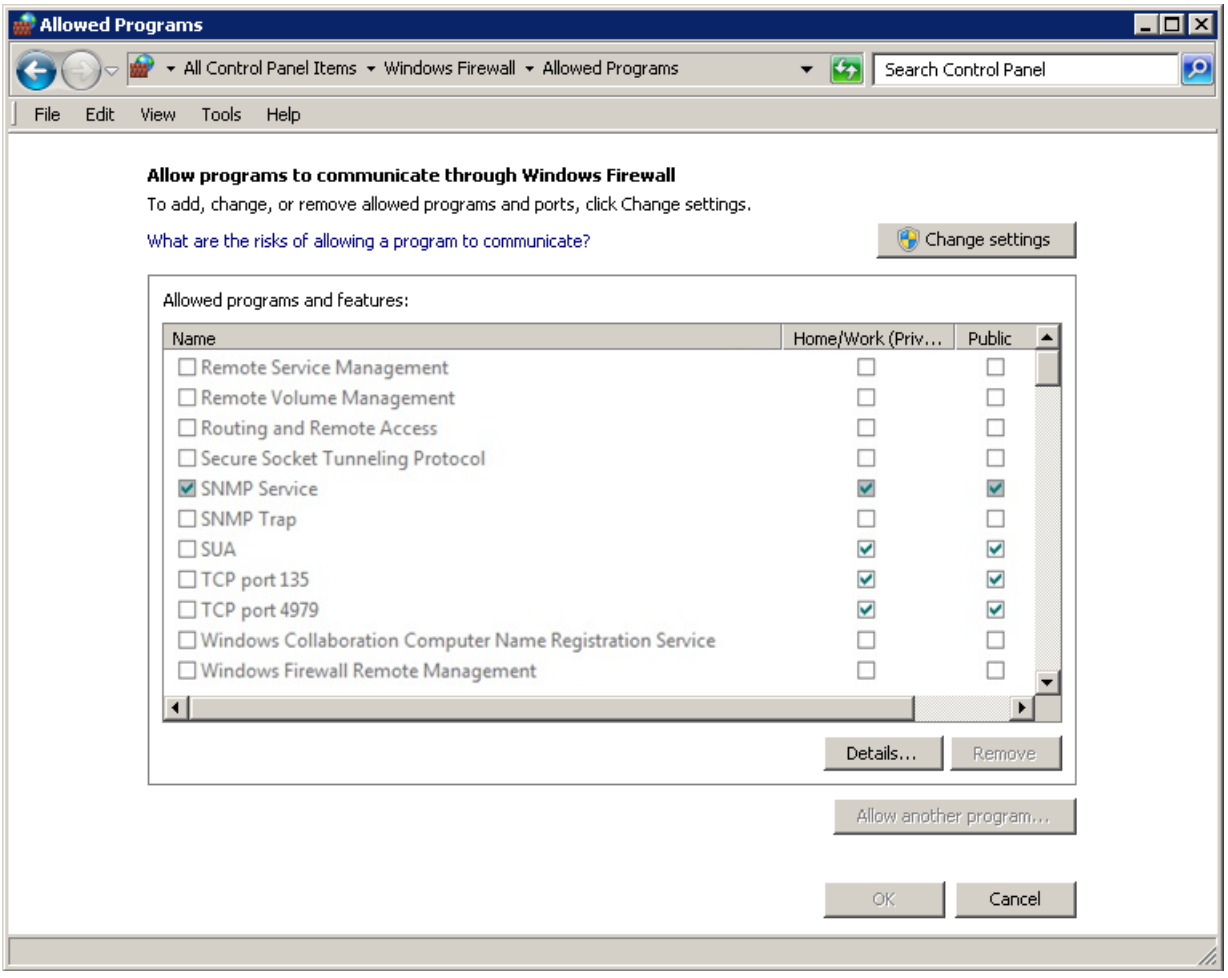
In the settings for the Windows Firewall, you can activate SNMP communication via the checkbox "SNMP Service" for specific network profiles.

### Procedure

To permit incoming SNMP communication on the Siemens PC system, proceed as follows:

1. Click "Start > Control Panel".
2. Enter "Windows Firewall" in the search field and then click on "Windows Firewall" in the search results.

3. Click "Allow a program or feature through Windows Firewall".
   The dialog box opens. The various checkboxes for "SNMP Service" are checked and displayed with a green or gray background, because the SNMP service is partially activated: Incoming queries are blocked on the Siemens PC system and outgoing queries from the PC system are permitted.



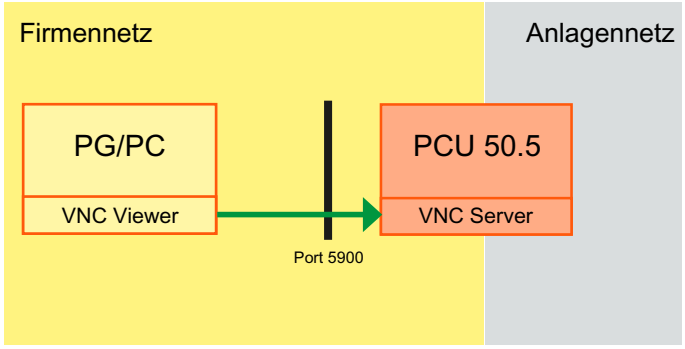

4. Click on "Change Settings" and confirm with "Yes", if applicable.
5. Activate the checkbox of the corresponding network profile completely, e.g. "Public" for the company network (factory setting).
6. Click "OK" to confirm the change.

## 4.17.7 Activate the remote access to the PC system

### 4.17.7.1 Overview

Basically, it is possible to access the Siemens PC system both via the plant network and via the company network via VNC:

- The remote access to the Siemens PC system from a PG/PC in the plant network is activated via the factory setting.
  See: Connection options in the system network (Page 141)
- If you want to access the Siemens PC system via the company network instead or in addition, you must adapt the firewall settings.




The inbound rule "vnc5900", which is deactivated in the factory settings state, is available for this purpose in the expanded settings for the Windows Firewall. To activate the remote access to the PC system in the company network, you can activate this rule either directly in the Control Panel (Page 72), or in the  prompt or by means of a script/batch file (Page 73).

#### Further information

In addition to the firewall settings, you may have to adapt other settings to establish a remote connection in the company network. You can find additional information on this in Chapter Remote access (Page 165), especially under Remote access to the PCU for operation and maintenance (Page 166).

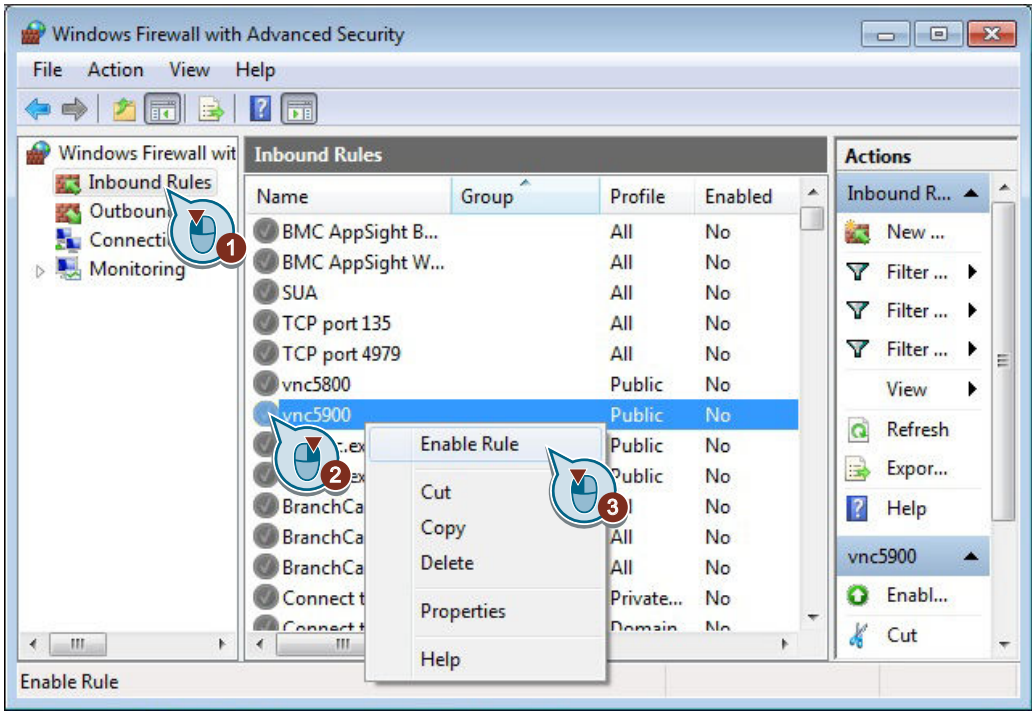
### 4.17.7.2 Via the Control Panel

#### Procedure

To activate remote access from the PG/PC in the company network to the Siemens PC system, proceed as follows:

1. Click "Start > Control Panel".
2. Enter "Windows Firewall" in the search field and then click on "Windows Firewall" in the search results.

3. Click on "Advanced settings" and confirm the call of the expanded settings with "Yes" if necessary.
4. In the "Windows Firewall with Advanced Security" dialog, right-click under "Inbound Rules" on the rule "vnc5900", then click on "Enable Rule".




## Result

With the activation of the inbound rule "vnc5900", Port 5900 was activated in the company network, which allows remote access from a PG/PC to the Siemens PC system in the company network.

If you want to deactivate remote access in the company network, right-click on the rule "vnc5900" and select "Disable Rule" in the shortcut menu.

## Further information

You can find general information on remote access to the Siemens PC system (via VNC Viewer or SSH client) in Chapter Remote access (Page 165).

### 4.17.7.3 By prompt, script or batch file

## Basic procedure

With a command according to the following scheme, you can activate remote access from the PG/PC in the company network to the Siemens PC system.

To do this, call the prompt as the administrator and enter the command either completely or successively.

Alternatively, you can save the command and the associated parameters in a script file (Page 69).

**Syntax for activating a firewall rule via the prompt**

```
netsh advfirewall firewall set rule name="<Name of the firewall rule>" new
enable=yes profile=<Name of the network profile>
```

Table 4-7 Description of the Netsh commands in the context of the advfirewall firewall

| Command | Parameter | Description | Value |
|---|---|---|---|
| netsh | | Specification of the command line program, which executes the following commands. | - |
| advfirewall | | Defines the context in which the following commands are to be carried out. | - |
| firewall | | Subcontext of "advfirewall". | - |
| set rule | | Adaptation of an existing firewall rule. | - |
| | name | Name of the firewall rule in quotation marks | "vnc5900" \| ... \| all |
| | new | Specifies that the following parameters are to be changed or added. | - |
| | enable | Activates or deactivates the specified firewall rule. | yes \| no |
| | profile | Name of the network profile for which the corresponding firewall rule is to be activated or deactivated. | public \| private \| domain \| any \| ... |

## Example

**Activation of the firewall rule "vnc5900" for the "Public" profile.**

```
netsh advfirewall firewall set rule name="vnc5900" new enable=yes profile=public
```

## Further information

Descriptions of the commands and of all of the associated parameters of Netsh within the Advfirewall firewall context can be found on the Microsoft website:

- Microsoft TechNet - Netsh AdvFirewall Firewall Commands (https://technet.microsoft.com/en-us/library/dd734783(v=ws.7).aspx#BKMK_3_set)
- Microsoft TechNet - Use of the command line tool Netsh AdvFirewall (https://technet.microsoft.com/en-us/library/cc770887(v=ws.7).aspx)

## 4.17.8 Activation/deactivation of the file and printer release

### 4.17.8.1 Overview

From a PG/PC in the plant network, you can easily and directly access a directory on the Siemens PC system.

If you want to access the PC system via the company network instead or in addition, you must adapt the firewall settings.

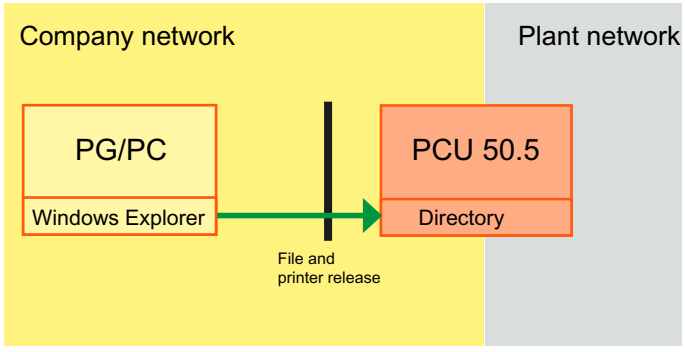



Figure 4-5 Access to a released directory of the PCU 50.5 from a PG/PC in the company network

### Overview

You can activate the file and printer release either for all of the network profiles, for a specific network profile, or for a specific network connection:

- General activation for all network profiles (Page 77)
  - Via the Control Panel (Page 77)
  - By prompt or script file (Page 78)
- Activation for a specific network profile (Page 80)
  - Via the Control Panel (Page 80)
  - By prompt or script file (Page 82)
- Activation for a specific connection (Page 85)
  - Via the Control Panel (Page 85)
  - By prompt or script file (Page 86)

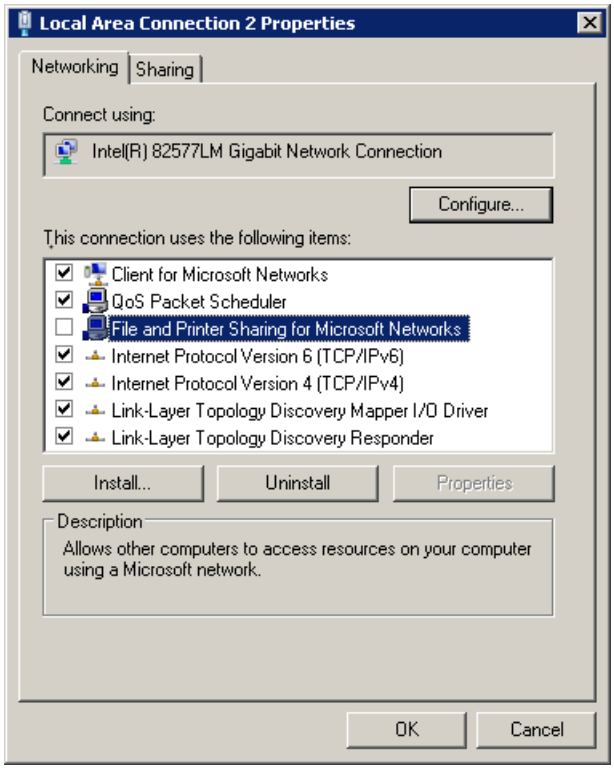
### 4.17.8.2 General information

#### General information on the file and printer release in Windows

In Windows, there are basically settings for the file and printer release at two different locations: In the settings of the corresponding network adapter and in the Windows Firewall settings.

- In the Windows Firewall settings, the file and printer release is at least partially blocked:
  - The Windows Firewall is deactivated in the plant network, however, which allows access to USB flash drives and external hard disks.
  - In the company network, the Windows Firewall is activated and the file and printer release is blocked by the Windows Firewall.

  These settings can be changed in various dialog boxes, e.g. in the "Network and Sharing Center" or in the "Allowed Programs" dialog box.
- In the settings of the network adapter for the plant network, the file and printer release is activated. However, in the settings of the network adapter for the company network, the file and printer release is deactivated. This setting primarily takes effect prior to the general setting of the Windows Firewall.




This can be seen in the corresponding Windows Firewall settings dialogs by means of a partially activated checkbox (green or gray background).



### Further information

- Microsoft Windows support - File and printer release: Frequently Asked Questions (http://windows.microsoft.com/en-us/windows/file-printer-sharing-faq#1TC=windows-7)

## 4.17.8.3 General activation for all network profiles

### Via the Control Panel

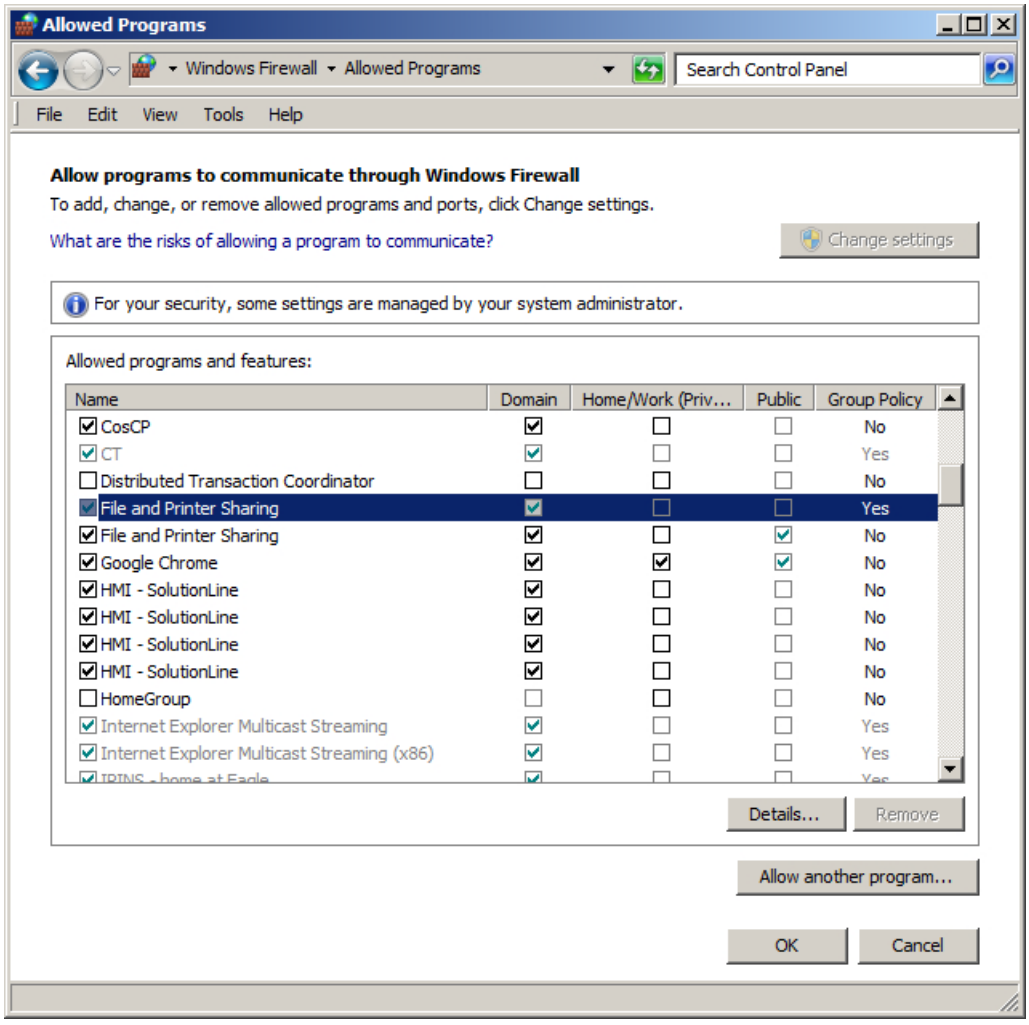
There are several procedures for activating the file and printer release for all of the network profiles in the Windows Firewall. (For example, there is the alternative procedure via the "Network Sharing Center" under "Change advanced sharing settings").

### Procedure

To activate the file and printer release on the Siemens PC system, proceed as follows:

1. Click "Start > Control Panel".
2. Enter "Windows Firewall" in the search field and then click on "Windows Firewall" in the search results.
3. Click "Allow a program or feature through Windows Firewall".

4. Click on "Change Settings" and confirm with "Yes", if applicable.
   The "Allowed Programs" dialog box opens.
   Also observe the entry in the column "Group Policy" and the color of the checkboxes (a checkbox with a green or gray background indicates partial activation).




5. Activate the corresponding checkbox for the corresponding entry "File and Printer Sharing" for the desired network profiles.
6. Click "OK" to confirm the change.

## By prompt, script or batch file

### Basic procedure

With a command according to the following scheme, you can allow the file and printer release in the company network for all of the network profiles together. All of the firewall rules of the group "File and Printer Sharing" are changed for all network profiles.

To do this, call the prompt as the administrator and enter the command either completely or successively.

Alternatively, you can save the command and the associated parameters in a script file (Page 69).

**Syntax for activating a group of firewall rules via the prompt**

```
netsh advfirewall firewall set rule group="<group name>" new enable=yes
```

Table 4-8 Description of the Netsh commands in the context of the advfirewall firewall

| Command | Parameter | Description | Value |
|---|---|---|---|
| netsh | | Specification of the command line program, which executes the following commands. | - |
| advfirewall | | Defines the context in which the following commands are to be carried out. | - |
| firewall | | Subcontext of "advfirewall". | - |
| set rule | | Adaptation of existing firewall rules. | - |
| | group | Name of the group with firewall rules | "File and Printer Sharing" \| ... |
| | new | Specifies that the following parameters are to be changed or added. | - |
| | enable | Activates or deactivates the specified group of firewall rules. | yes \| no |

## Example

**Activation of the firewall rule "File and printer release" for the "Public" profile.**

```
netsh advfirewall firewall set rule group="File and Printer Sharing" new enable=yes
```

## Further information

Descriptions of the commands and of all of the associated parameters of Netsh within the Advfirewall firewall context can be found on the Microsoft website:

- Microsoft TechNet - Netsh AdvFirewall Firewall Commands (https://technet.microsoft.com/en-us/library/dd734783(v=ws.7).aspx#BKMK_3_set)

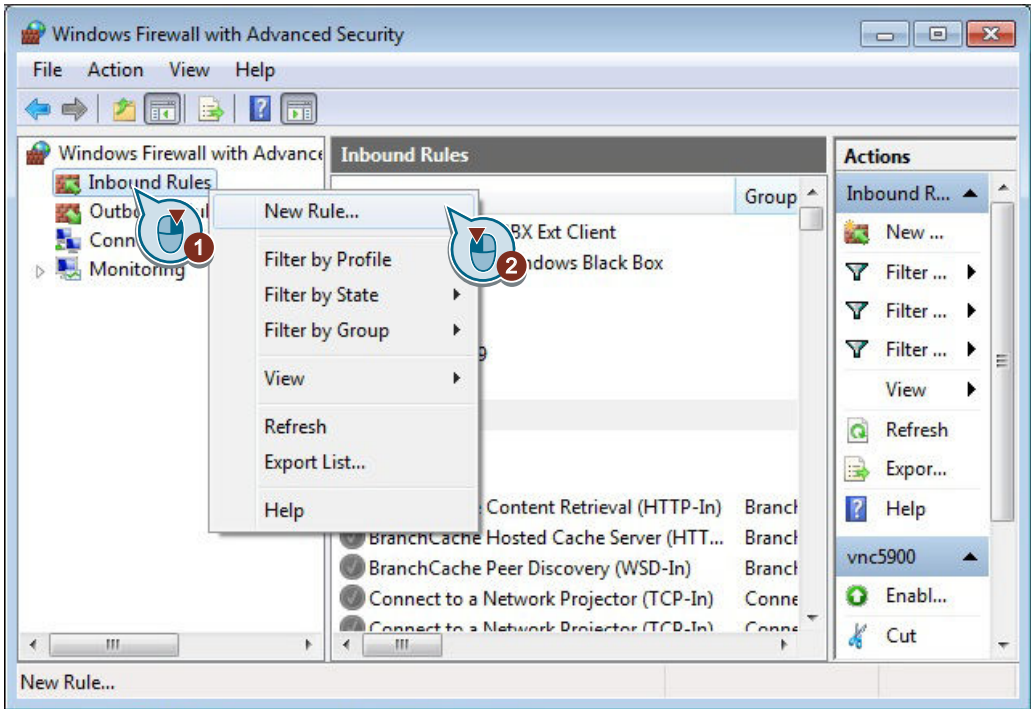
### 4.17.8.4 Activation for a specific network profile

## Via the Control Panel

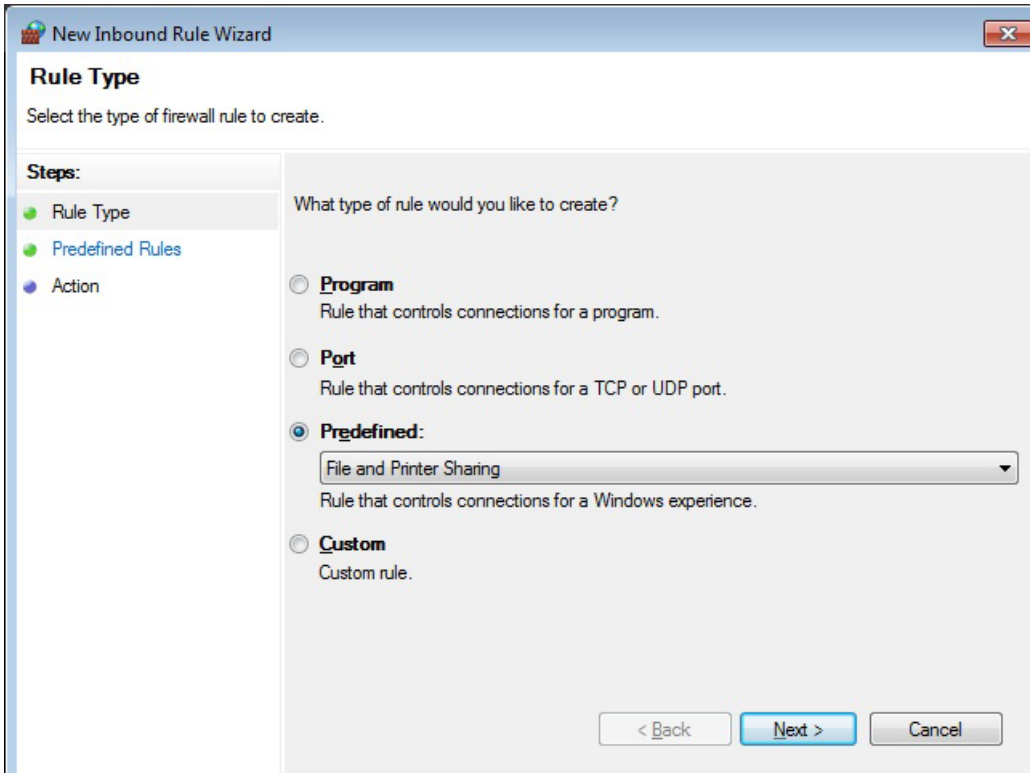
## Procedure

To activate the file and printer release from a PG/PC in the company network to PCU 50.5 for a specific network profile, proceed as follows:

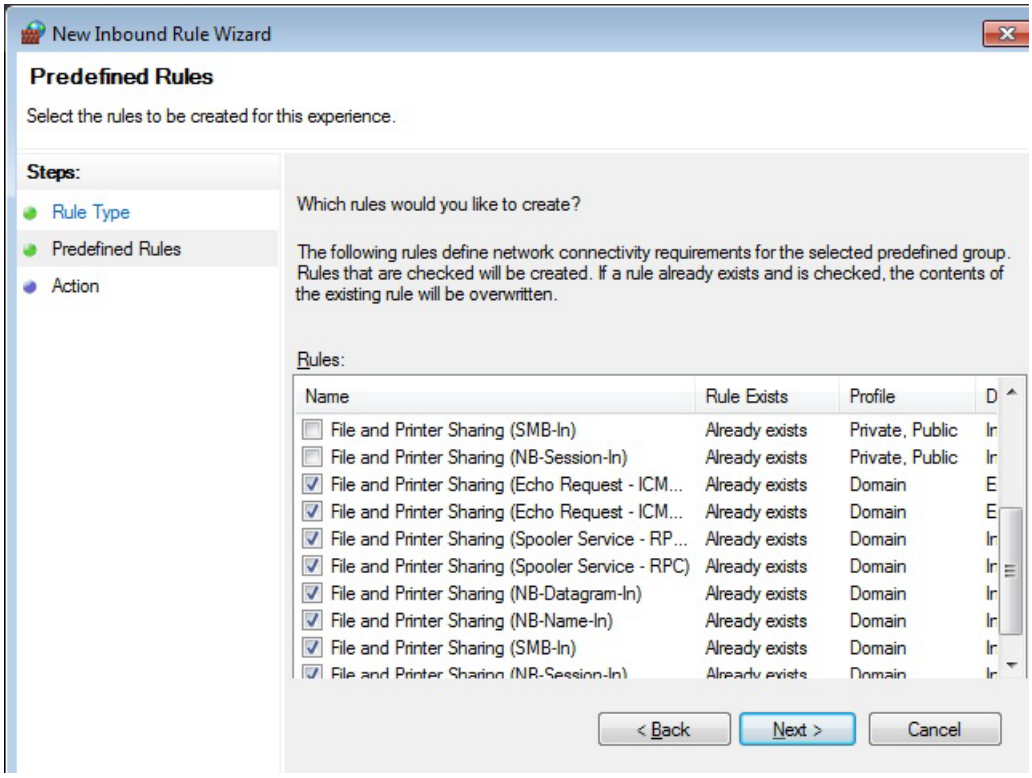
1. Click "Start > Control Panel".
2. Enter "Windows Firewall" in the search field and then click on "Windows Firewall" in the search results.
3. Click on "Advanced settings" and confirm the call of the expanded settings with "Yes" if necessary.
4. In the "Windows Firewall with Advanced Security" dialog, right-click on the rule "Inbound Rules", then click on "New Rule..."



5. In the "New Inbound Rule Wizard" dialog, select the option button "Predefined", select in the "File and Printer Sharing" drop-down list, and click on "Next".



6. Depending on the network profile (e.g. Public or Domain) for which the file and printer release is to be activated, check the corresponding checkbox.




7. Click "Next".
8. Select the option button "Allow the connection" and click on "Finish".

## By prompt, script or batch file

### Basic procedure

With a command according to the following scheme, you can switch off the blocking of the file and printer release by the Windows Firewall for a special network profile.

To do this, call the prompt as the administrator and enter the command either completely or successively.

Alternatively, you can save the command and the associated parameters in a script file (Page 69).

**Syntax for the activation of all firewall rules regarding file and printer release**

```
netsh
advfirewall firewall
set rule name="File and Printer Sharing (NB-Session-In)" new enable=yes
profile=<profile>
set rule name="File and Printer Sharing (NB-Session-Out)" new enable=yes
profile=<profile>
set rule name="File and Printer Sharing (SMB-In)" new enable=yes
profile=<profile>
set rule name="File and Printer Sharing (SMB-Out)" new enable=yes
profile=<profile>
set rule name="File and Printer Sharing (NB-Name-In)" new enable=yes
profile=<profile>
set rule name="File and Printer Sharing (NB-Name-Out)" new enable=yes
profile=<profile>
set rule name="File and Printer Sharing (NB-Datagram-In)" new enable=yes
profile=<profile>
set rule name="File and Printer Sharing (NB-Datagram-Out)" new enable=yes
profile=<profile>
set rule name="File and Printer Sharing (Spooler Service - RPC)" new
enable=yes profile=<profile>
set rule name="File and Printer Sharing (Spooler Service - RPC-EPMAP)" new
enable=yes profile=<profile>
set rule name="File and Printer Sharing (Echo Request - ICMPv4-In)" new
enable=yes profile=<profile>
set rule name="File and Printer Sharing (Echo Request - ICMPv4-Out)" new
enable=yes profile=<profile>
set rule name="File and Printer Sharing (Echo Request - ICMPv6-In)" new
enable=yes profile=<profile>
set rule name="File and Printer Sharing (Echo Request - ICMPv6-Out)" new
enable=yes profile=<profile>
set rule name="File and Printer Sharing (LLMNR-UDP-In)" new enable=yes
profile=<profile>
set rule name="File and Printer Sharing (LLMNR-UDP-Out)" new enable=yes
profile=<profile>
```

Table 4-9 Description of the Netsh commands in the context of the advfirewall firewall

| Command | Parameter | Description | Value |
|---|---|---|---|
| netsh | | Specification of the command line program, which executes the following commands. | - |
| advfirewall | | Defines the context in which the following commands are to be carried out. | - |
| firewall | | Subcontext of "advfirewall". | - |
| set rule | | Adaptation of existing firewall rules. | - |
| | name | Name of the firewall rule in quotation marks. | "File and Printer Sharing (NB-Session-In)" \| ... |

| Com-mand | Parameter | Description | Value |
|---|---|---|---|
| | new | Specifies that the following parameters are to be changed or added. | - |
| | enable | Activates or deactivates the specified firewall rule. | yes \| no |

## Example

**Example for the activation of all firewall rules regarding file and printer release**

```
netsh
advfirewall firewall
set rule name="File and Printer Sharing (NB-Session-In)" new enable=yes
profile=public
set rule name="File and Printer Sharing (NB-Session-Out)" new enable=yes
profile=public
set rule name="File and Printer Sharing (SMB-In)" new enable=yes
profile=public
set rule name="File and Printer Sharing (SMB-Out)" new enable=yes
profile=public
set rule name="File and Printer Sharing (NB-Name-In)" new enable=yes
profile=public
set rule name="File and Printer Sharing (NB-Name-Out)" new enable=yes
profile=public
set rule name="File and Printer Sharing (NB-Datagram-In)" new enable=yes
profile=public
set rule name="File and Printer Sharing (NB-Datagram-Out)" new enable=yes
profile=public
set rule name="File and Printer Sharing (Spooler Service - RPC)" new
enable=yes profile=public
set rule name="File and Printer Sharing (Spooler Service - RPC-EPMAP)" new
enable=yes profile=public
set rule name="File and Printer Sharing (Echo Request - ICMPv4-In)" new
enable=yes profile=public
set rule name="File and Printer Sharing (Echo Request - ICMPv4-Out)" new
enable=yes profile=public
set rule name="File and Printer Sharing (Echo Request - ICMPv6-In)" new
enable=yes profile=public
set rule name="File and Printer Sharing (Echo Request - ICMPv6-Out)" new
enable=yes profile=public
set rule name="File and Printer Sharing (LLMNR-UDP-In)" new enable=yes
profile=public
set rule name="File and Printer Sharing (LLMNR-UDP-Out)" new enable=yes
profile=public
```

## Further information

Descriptions of the commands and of all of the associated parameters of Netsh within the Advfirewall firewall context can be found on the Microsoft website:

- Microsoft TechNet - Netsh AdvFirewall Firewall Commands (https://technet.microsoft.com/en-us/library/dd734783(v=ws.7).aspx#BKMK_3_set)

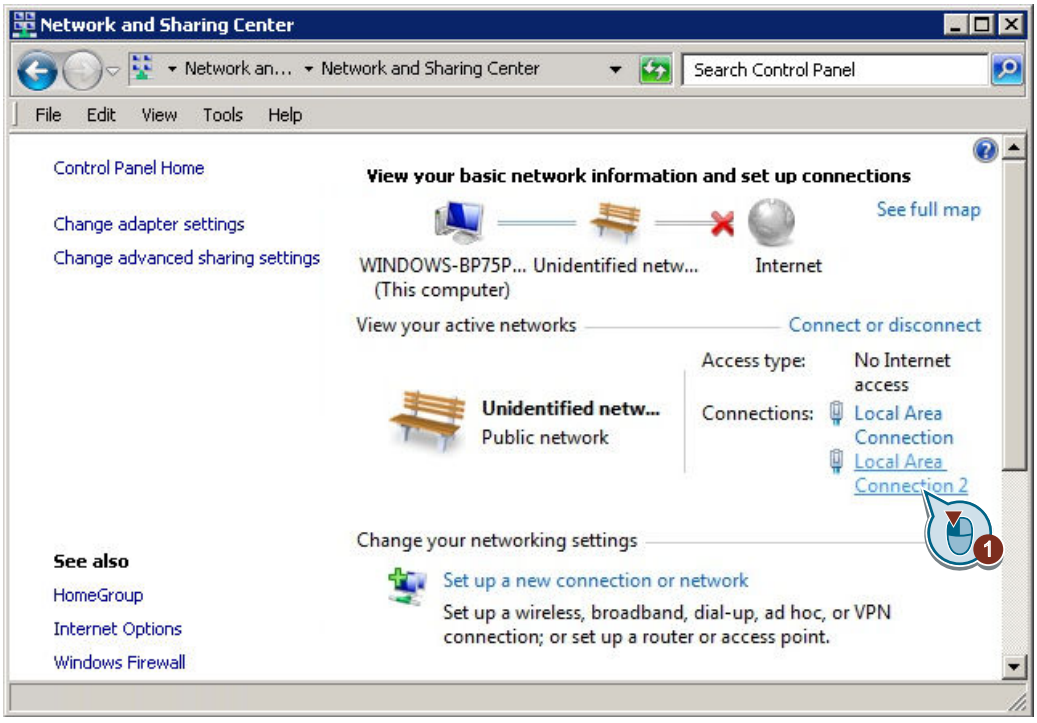
### 4.17.8.5 Activation for a specific connection

#### Via the Control Panel

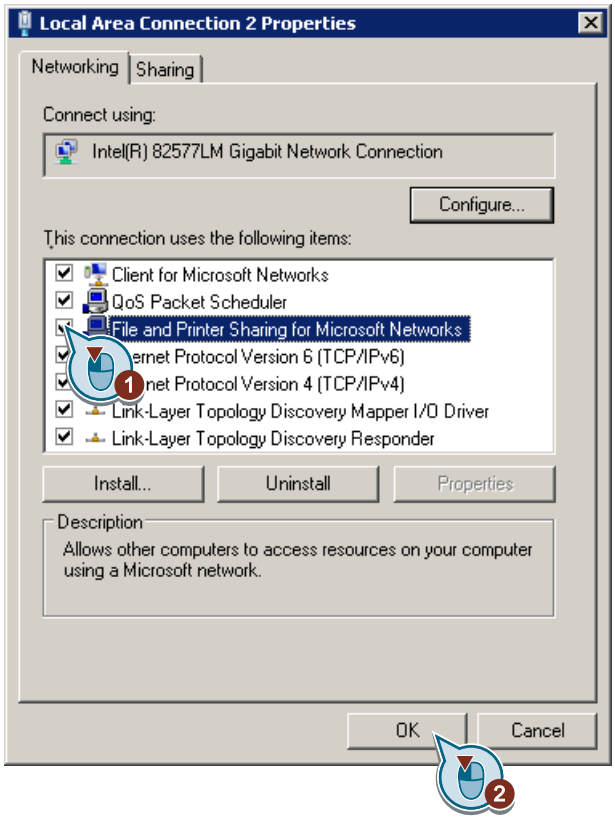
#### Procedure

To activate the file and printer release for a specific connection (e.g. company network), proceed as follows:

1. Click "Start > Control Panel".
2. Enter "Network and Sharing Center" in the search field and then click on "Network and Sharing Center" in the search results.
3. Click on the desired connection, e.g. "Local Area Connection 2" (company network).



4. Click on "Properties" and confirm the call of the properties with "Yes" if applicable.
5. In the "Networking" tab, check the checkbox "File and Printer Sharing for Microsoft Networks" and confirm the change with "OK".




## By prompt, script or batch file

In Windows, the activation of the file and printer release for a specific connection via the prompt or script/batch file is not supported. This is possible, however, with the aid of the separate, free command line program "Hyper-V Network VSP Bind (nvspbind)":

- Microsoft TechNet - Hyper-V Network VSP Bind (nvspbind) (https://gallery.technet.microsoft.com/Hyper-V-Network-VSP-Bind-cf937850)
  Command line program for configuring network adapters and connections via the prompt

You can find additional information in the documentation for nvspbind.

**Example: Activation of the file and printer release for "Local Area Connection 2" (company network)**

```
nvspbind -e "Local Area Connection 2" ms_server
```

## 4.17.9 Activation of the ping execution (ICMP)

### 4.17.9.1 Via the Control Panel

In the factory setting state, incoming ping requests are blocked on the company network interface (X1) of the Siemens PC system. Outgoing ping requests are permitted.

**Note**

**Dependency for the file and printer release**

If you activated the file and printer release in the company network, incoming ping requests are already permitted because they are required for the file and printer release.

### Procedure

To permit ICMP pings, which arrive at the company network interface (X1) of the Siemens PC system, proceed as follows:

1. Click "Start > Control Panel".
2. Enter "Windows Firewall" in the search field and then click on "Windows Firewall" in the search results.
3. Click on "Advanced settings" and confirm the call of the expanded settings with "Yes" if necessary.
4. In the "Windows Firewall with Advanced Security" dialog, click on "Inbound Rules".

5. Select the corresponding rule for the relevant network profile:
   – File and Printer Sharing (Echo Request - ICMPv4-In) - Profile "Domain"
   – File and Printer Sharing (Echo Request - ICMPv4-In) - Profile "Public"
   – File and Printer Sharing (Echo Request - ICMPv4-In) - Profile "Private"

   In the factory setting state, the network profile "Public" is assigned to the company network.

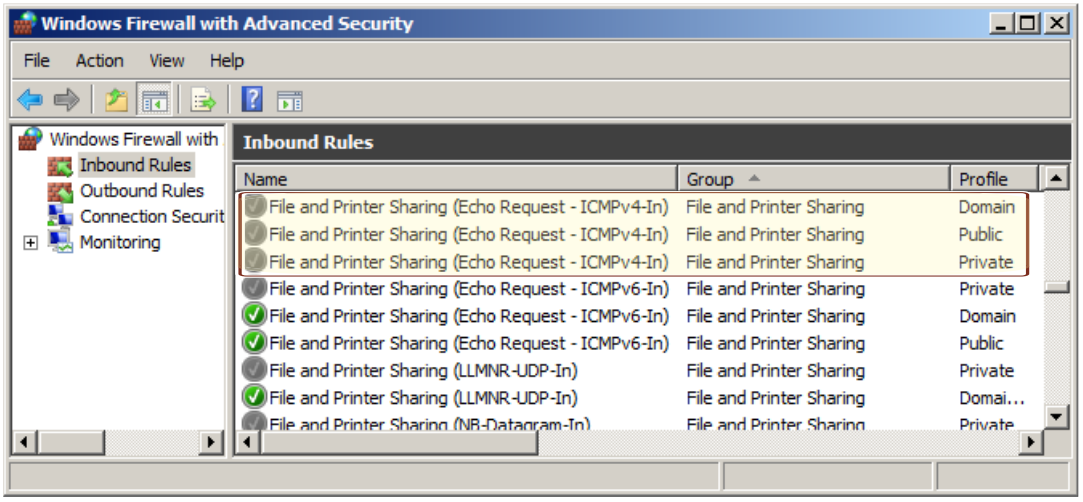



Figure 4-6 Rules for incoming ICMP pings, separate for each network profile

6. Right-click on the selected rule, then click "Enable Rule".

### 4.17.9.2 By prompt, script or batch file

#### Basic procedure

With a command according to the following scheme, you can permit ICMP pings which arrive at the company network interface (X1) of the Siemens PC system. Outgoing pings from the Siemens PC system are already activated.

**Note**

**Dependency for the file and printer release**

If you activated the file and printer release in the company network, incoming ping requests are already permitted because they are required for the file and printer release.

The name of the relevant firewall rule is "File and Printer Sharing (Echo Request - ICMPv4-In)". To do this, call the prompt as the administrator and enter the command either completely or successively.

Alternatively, you can save the command and the associated parameters in a script or batch file (Page 69).

**Syntax for activating a firewall rule via the prompt**

```
netsh advfirewall firewall set rule name="<Name of the firewall rule>" new enable=yes
profile=<Name of the network profile>
```

Table 4-10 Description of the Netsh commands in the context of the advfirewall firewall

| Command | Parameter | Description | Value |
|---|---|---|---|
| netsh | | Specification of the command line program, which executes the following commands. | - |
| advfirewall | | Defines the context in which the following commands are to be carried out. | - |
| firewall | | Subcontext of "advfirewall". | - |
| set rule | | Adaptation of existing firewall rules. | - |
| | name | Name firewall rule | "File and Printer Sharing (Echo Request - ICMPv4-In)" \| ... |
| | new | Specifies that the following parameters are to be changed or added. | - |
| | enable | Activates or deactivates the specified firewall rule. | yes \| no |
| | profile | Name of the network profile for which the firewall rule is to be changed. | |

### Example

**Activation of incoming ping requests**

```
netsh advfirewall firewall set rule name="File and Printer Sharing (Echo Request - ICMPv4-
In)" new enable=yes profile=public
```

### Further information

Descriptions of the commands and of all of the associated parameters of Netsh within the Advfirewall firewall context can be found on the Microsoft website:

- Microsoft TechNet - Netsh AdvFirewall Firewall Commands (https://technet.microsoft.com/en-us/library/dd734783(v=ws.7).aspx#BKMK_3_set)

## 4.17.10 Saving and restoring firewall settings

### 4.17.10.1 Overview

### Overview

You can save all of the Windows Firewall settings of the Siemens PC system as a policy file with the file extension ".wfw". Policy files are saved in binary format and are Windows version-dependent.

You have the following options for backing up or restoring the firewall settings:

- Backup of firewall settings (Page 90)
  - Via the Control Panel (Page 90)
  - By prompt, script or batch file (Page 90)
- Restoring of firewall settings (Page 91)
  You can restore the firewall settings on the same Siemens PC system or import the same policy file to different Siemens PC systems for the purpose of series commissioning.
  - Via the Control Panel (Page 91)
  - By prompt, script or batch file (Page 92)

### 4.17.10.2 Backup of firewall settings

#### Via the Control Panel

#### Procedure

To back up the Windows Firewall settings of the Siemens PC, proceed as follows:

1. Click "Start > Control Panel".
2. Enter "Windows Firewall" in the search field and then click on "Windows Firewall" in the search results.
3. Click on "Advanced settings" and confirm the call of the expanded settings with "Yes" if necessary.
4. Right-click on the master object "Windows Firewall with Advanced Security on Local Computer" and then click on "Export Policy...".
5. In the "Save As" dialog, select a save location and file name and confirm with "Save".

#### By prompt, script or batch file

#### Basic procedure

With a command according to the following scheme, you can back up the Windows Firewall settings in a policy file (.wfw).

To do this, call the prompt as the administrator and first define the save location of the policy memory, then enter a file path for backing up the policy file.

Alternatively, you can save the commands and the associated parameters in a script or batch file (Page 69).

**Syntax of the Netsh commands for defining the policy memory and for exporting the policy file**

```
netsh advfirewall set store local
export <File path and file name>.wfw
```

Table 4-11 Description of the Netsh commands for defining the policy memory and for exporting the policy file

| Command | Parameter | Description | Value |
|---|---|---|---|
| netsh | | Specification of the command line program, which executes the following commands. | - |
| advfirewall | | Defines the context in which the following commands are to be carried out. | - |
| set store | | Definition of the save location of the policy memory for the following commands. | - |
| | local | Defines that the work is to be done with the local policy memory. | - |
| export | | Backup of all firewall settings in a policy file (.wfw). | <File path and file name>.wfw |

### Example

**Example of the Netsh commands for defining the policy memory and for exporting the policy file**

```
netsh advfirewall set store local
export D:\Backup\DefaultSettings.wfw
```

### Further information

Descriptions of the commands and of all of the associated parameters of Netsh within the Advfirewall context can be found on the Microsoft website:

- Microsoft TechNet: Netsh Commands for Windows Firewall with Advanced Security (https://technet.microsoft.com/en-us/library/cc771920(v=ws.7).aspx)

## 4.17.10.3 Restoring of firewall settings

### Via the Control Panel

### Procedure

**Note**

**Importing a policy file overwrites all existing firewall rules**

When the firewall settings are restored, all of the existing firewall rules are overwritten. You cannot undo the import of firewall rules via a policy file.

Therefore, back up (export) the existing firewall settings before you import a policy file.

To restore the Windows Firewall settings of the Siemens PC system, proceed as follows:

1. Click "Start > Control Panel".
2. Enter "Windows Firewall" in the search field and then click on "Windows Firewall" in the search results.
3. Click on "Advanced settings" and confirm the call of the expanded settings with "Yes" if necessary.
4. Right-click on the master object, then click "Windows Firewall with Advanced Security on Local Computer", then click on "Import Policy..." and confirm the command with "Yes".
5. In the "Open" dialog, select the policy file and confirm with "Open".

## By prompt, script or batch file

### Basic procedure

**Note**

**Importing a policy file overwrites all existing firewall rules**

When the firewall settings are restored, all of the existing firewall rules are overwritten. You cannot undo the import of firewall rules via a policy file.

Therefore, back up (export) the existing firewall settings before you import a policy file.

With a command according to the following scheme, you can restore the Windows Firewall settings from a policy file (.wfw).

To do this, call the prompt as the administrator and first define the save location of the policy memory, then enter a file path for backing up the policy file.

Alternatively, you can save the commands and the associated parameters in a script or batch file (Page 69).

**Syntax of the Netsh commands for defining the policy memory and for importing the policy file**

```
netsh advfirewall set store local
import <File path and file name>.wfw
```

Table 4-12 Description of the Netsh commands for defining the policy memory and for importing the policy file

| Command | Parameter | Description | Value |
|---|---|---|---|
| netsh | | Specification of the command line program, which executes the following commands. | - |
| advfirewall | | Defines the context in which the following commands are to be carried out. | - |
| set store | | Definition of the save location of the policy memory for the following commands. | - |

| Com-mand | Parameter | Description | Value |
|---|---|---|---|
| | local | Defines that the work is to be done with the local policy memory. | - |
| import | | Restoration of firewall settings from a policy file (.wfw). | <File path and file name>.wfw |

**Example of the Netsh commands for defining the policy memory and for importing the policy file**

```
netsh advfirewall set store local
import D:\Backup\DefaultSettings.wfw
```

## Further information

Descriptions of the commands and of all of the associated parameters of Netsh within the Advfirewall context can be found on the Microsoft website:

- Microsoft TechNet: Netsh Commands for Windows Firewall with Advanced Security (https://technet.microsoft.com/en-us/library/cc771920(v=ws.7).aspx)

# Installing and configuring updates and automation software

# 5

## 5.1 Overview

Once you have completed installation of the PCU-Basesoftware and possibly configured the network settings, you can set up handling of the security updates, patches, and software installations or install them directly.

- You will find information about Windows updates and the network security of an automation system in Chapter Patch management and security updates (Page 96).
- The PCU Installer (Page 108) will assist you with automatic installation of setup packages.
- You can install SINUMERIK Operate (Page 98) or other HMI software and at the same time (Page 98) orsubsequently (Page 100) set it up for autologon mode.
- To prevent operation of the HMI software from being interrupted by key combinations such as CTRL+ALT+DEL, you can activate and configure the key filter (Page 102).
- You can install STEP 7 and the SINUMERIK add-on (Page 106) and add a softkey to the HMI software to start STEP 7 (Page 107).
- If you want to transfer existing data to the newly installed automation software, read ChapterMigration (Page 119).

## 5.2 Patch management and security updates

### Overview

Microsoft regularly eliminates security holes in Windows and makes these corrections available on its "Windows Update" website.

As a solution for managing and making Windows updates available in a network, Microsoft offers the component Windows Server Update Services (WSUS).

Microsoft TechNet - Windows Server Update Services (http://technet.microsoft.com/en-us/wsus/default.aspx)

#### Industrial security

You will find information about network security for an automation system and a comprehensive protection concept under Industrial Security on the SIEMENS website:

SIEMENS Industrial Security (http://www.industry.siemens.com/topics/global/en/industrial-security/)

SINUMERIK-specific information about Industrial Security can also be found in the following documents:

SINUMERIK/SIMOTION/SINAMICS Configuration Manual Industrial Security (https://support.industry.siemens.com/cs/document/108862708?lc=en-WW)

#### Compatibility with automation software

Information about compatibility of a Windows update with specific application software of SIEMENS may possibly be available on the product pages of the application software in question in the Service&Support portal:

SIEMENS Industry Service & Support Portal (https://support.industry.siemens.com/)

Generally, it is recommended that you check the compatibility of updates in a dedicated project test environment before installing it in a productive environment.

## 5.3 Configuring saving Windows log files

As a result of Windows updates and other operations, larger quantities of data can accumulate that are no longer required, e.g. temporary update and system files or Windows log files (`CBS.log`).

The PCU base software contains 2 batch files, which you can use to configure how files such as these are saved:

- Deactivate saving
  - Saving new log files `CBS.log` is deactivated, and existing files are deleted: `C:\Windows\Logs\CBS`
  - The temporary directory is cleared once: C:\Windows\Temp
  - The data storage medium cleanup is carried out.
- Activate saving
  - Reactivates that new log files `CBS.log` are saved for diagnostics.

You can refer to the information file in the following directory to identify whether the log files are being saved or not: `C:\Windows\Logs\CBS`

### Precondition

- You require administrator rights to run batch files.

### Procedure

1. Change to directory `D:\etc` and double-click on the appropriate batch file.

## 5.4 Setting up SINUMERIK Operate for autologon mode

If you want the PC system to start automatically in SINUMERIK Operate after switching on without having to make any further entries, set up the Autologon mode.

During installation of SINUMERIK Operate, you can define an existing user account as an autologon account. The PC system then starts without the password for this user account being entered. SINUMERIK Operate is then started automatically because a link is created in the "Startup" Autostart folder.

For security reasons, you should use a special user account without administrator rights.

You can also configure the key filter and deactivate certain keyboard entries (Page 102), e.g. to prevent a switchover from SINUMERIK Operate to the Windows Desktop. (Desktop access is then only possible via the splash screen (Page 160).)

### Requirement

- A Windows user account without administrator rights is set up.
- SINUMERIK Operate is not installed.
- You are logged on as an administrator or you know the password of the administrator account.
- The keyboard layout of the system is set to English.
  See: Configuring the keyboard layout (Page 35)

## Procedure

To set up autologon mode, proceed as follows:

1. Start installation of SINUMERIK Operate and, in the installation wizard, proceed to step "Autologon".

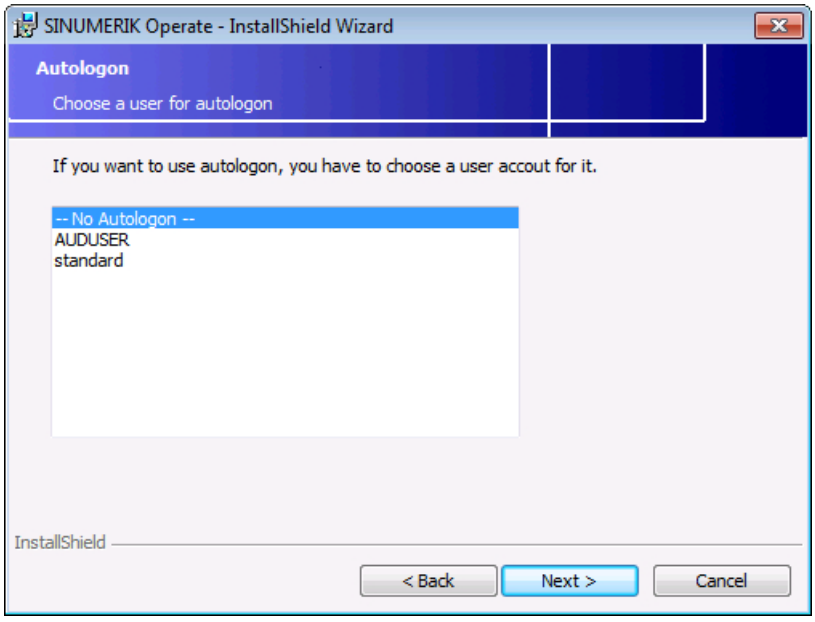



Figure 5-1 SINUMERIK Operate Installer - Autologon

2. In the list, select the user account to be logged on automatically.
3. Confirm with "Next", then in the next steps of the installation wizard make all the necessary settings and confirm the installation with "Finish".

## Result

SINUMERIK Operate is installed and Autologon mode of the PC system is set up. During this process the password of the autologon user account is stored encrypted in Windows, and SINUMERIK Operate is linked with the autostart folder "Startup".

Autologon mode is active at the next startup. The autologon user account is automatically logged on and SINUMERIK Operate is started.

## 5.5 Modifying the user account for autologon mode.

Because of the way autologon works, only one user account can ever be set up for autologon mode.

If you set up autologon mode during installation of SINUMERIK Operate for one particular user account, but now want to use another user account for autologon mode, you can change this manually in the Service Desktop.

### Requirement

SINUMERIK Operate is installed and autologon mode is set up.

The Service Desktop is active.

## Procedure

To change the user account that is used for automatic startup in autologon mode, proceed as follows:

1. To open the "User Accounts" dialog box, proceed in one of the following ways:
   - On the Desktop, double-click the "Netplwiz" icon.
   - In the search field of the Start menu, enter `netplwiz` and confirm the entry.
   - At the prompt, enter `control userpasswords2`.

   The "User Accounts" dialog box opens.

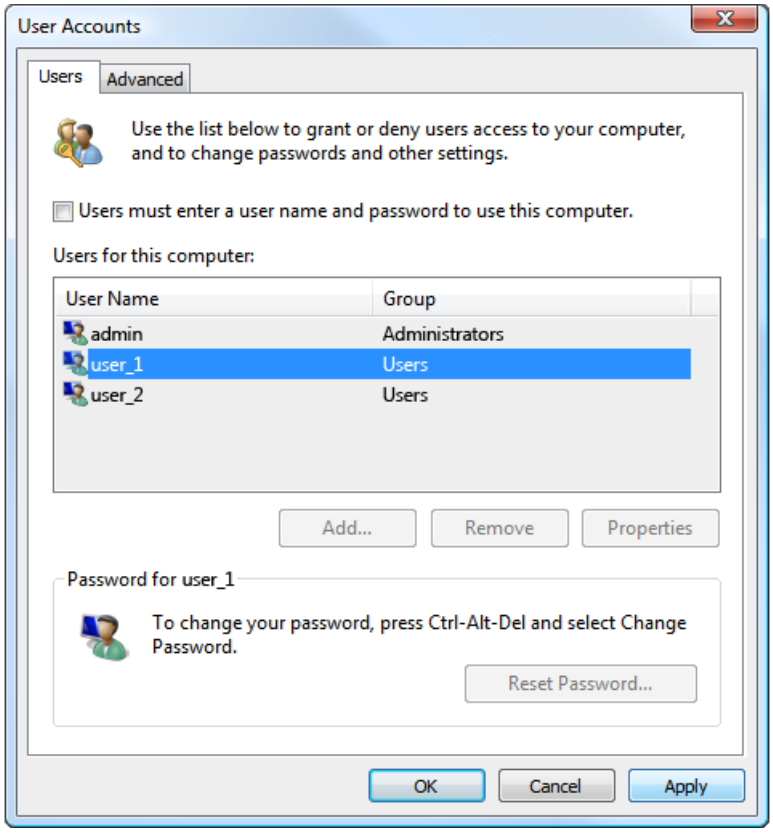



Figure 5-2 "User Accounts" dialog box in Windows 7: Please note that the deactivated checkbox refers to the selected user account.

2. Activate the "Users must enter a user name and password to use this computer" checkbox.
3. In the "Users for this computer" list, select the user account to be used for autologon mode.
4. Deactivate the "Users must enter a user name and password to use this computer" checkbox.
   You may be prompted to enter a password.
5. Confirm these settings in the "User Accounts" dialog box with "OK".
6. In the Autostart folder of the autologon mode user account, create a link to SINUMERIK Operate.

## 5.6 Configuring a key filter for HMI software

To prevent unwanted interruption of the HMI software while it is running, you can specify that certain keyboard entries will be filtered out and thus ignored.

You can configure the key filter in the pcuhwsvc.ini configuration file:

- This key filter is not activated as delivered.
- You can activate the key filter and individually select which of the 20 filterable keyboard entries will be ignored. During this, you can deactivate known shortcut keys, for example, or even deactivate individual add-on keys, such as the left (SeqActive_16) or right (SeqActive_17) Windows key.
  You cannot configure any additional keyboard entries to be filtered.
- You can also completely deactivate the repeat function of the F1-F12 keys and all hard keys with the key filter. When the repeat function is deactivated, a key held pressed is considered as pressed once and not as pressed repeatedly.
  Per default, the repeat function is activated during Service or Desktop operation. The repeat function is always deactivated during operation of SINUMERIK Operate.

### Requirement

- The Windows 7 Service Desktop is active
- Hidden directories are shown
  See also: Directory structure and file conventions (Page 14).

### Procedure

To configure the key filter, proceed as follows:

1. Switch to the template directory: `C:\ProgramData\Siemens\MotionControl\siemens\System\etc\`
   The settings can be viewed in the template `pcuhwsvc.ini`.
2. Save a copy of the template `pcuhwsvc.ini` to one of the user directories:
   - `C:\ProgramData\Siemens\MotionControl\user\System\etc\`
   - `C:\ProgramData\Siemens\MotionControl\oem\System\etc\`
   - `C:\ProgramData\Siemens\MotionControl\addon\System\etc\`
3. In the new file (e.g. `...\oem\System\etc\pcuhwsvc.ini`), delete all data except for the section name `[KEYB_FILTER]` and the keys that are to differ from the template.
   All data and keys that do not occur in your file are automatically taken over from the template in the `siemens` directory.
   See also: Directory structure and file conventions (Page 14).
4. Save your file.

### Example: Activating the key filter, but allowing use of the NUM-lock key.

This example activates the key filter (`KeySequencesEnable = 1`) and filters all possible keyboard entries except the NUM-lock key.

| | |
|---|---|
| `KEYB_FILTER` | Section name (mandatory): |
| `KeySequencesEnable = 1` | Activate key filter |
| `SeqActive_19= 0` | Do not filter NUM-lock key |

In the adapted file in the user directory (e.g. `...\oem\System\etc\`) therefore only the section name `[KEYB_FILTER]` is mandatory as well as the settings that you want to adapt. All other settings and values are taken over from the template.

### Template of the key filter in the delivery condition

The template is located in directory `C:\ProgramData\Siemens\MotionControl\siemens\System\etc\`

Do not overwrite this template, but save a copy of this template `pcuhwsvc.ini` in one of the user directories.

- `C:\ProgramData\Siemens\MotionControl\`**user**`\System\etc\`
- `C:\ProgramData\Siemens\MotionControl\`**oem**`\System\etc\`
- `C:\ProgramData\Siemens\MotionControl\`**addon**`\System\etc\`

See also: Directory structure and file conventions (Page 14).

#### How the configuration file works

You can configure the key filter in section `[KEYB_FILTER]`. The lines identified by a number sign (#) are comments as to which line further down deactivates a particular keyboard entry.

- To activate a particular keyboard entry, find the key in the comment and then set the value to 0.
- To allow all keyboard entries and thus deactivate the key filter (factory setting), set the value to 0 in the `KeySequencesEnable` key.
  If the value 0 is set for this key, all other keys will be ignored and the key filter is deactivated.
- To completely deactivate the repeat function of the F1-F12 keys and all hard keys, set the value 1 at the `AutorepeatSuppression` key.

Table 5-1 Overview: Key filter in the configuration file pcuhwsvc.ini

| | |
|---|---|
| Section | [KEYB_FILTER] |
| Key | SeqActive_x |
| Value | 0 = specific keyboard entry activated<br>1 = specific keyboard entry will be filtered and is deactivated |
| Factory setting | 1 |

**pcuhwsvc.ini (Vorlage im Verzeichnis ...\siemens\System\etc)**

```
[GLOBAL]
# ----------------------
```

**pcuhwsvc.ini (Vorlage im Verzeichnis ...\siemens\System\etc)**

```
# SIEMENS GLOBAL SECTION
# ----------------------
# enable logfile PCUHardwareservice.log
EnableLogFile=0
#  Emulating of "Mode-Switch" from Sinumerik PCU50 Box.
# "Mode-Switch" is a rotary switch, which is attached to the rear side of the PCU50 V5 Box.
#  About this, different methods of booting the computer are controlled.
# 0  Normal mode
# 3  Desktop mode (welcome screen)
ModeSwitch=0
[SERVER]
FTP=hmisvr_PCU_betaftpd
DHCP=hmisvr_PCU_udhcpd
TFTP=hmisvr_PCU_netkit-tftpd
TCUHWS=hmisvr_TCU_hardware_services
VNC=uvnc_service

[APPLICATION]
# ---------------------------------------
# STARTUP APPLICATION DIAGNOSTIC SECTION
# ---------------------------------------

[KEYB_FILTER]
# -----------------------------------------
# SIEMENS KEYBOARD-FILTER DRIVER SECTION
# -----------------------------------------

# Filtering Key Sequences
# =======================
# Activation of Key-Sequences to be ignored
# SeqAct_x
# The following 20 sequences are implemented
# 0  CTRL-ALT-DEL
# 1  ALT-F4
# 2  ALT-TAB
# 3  LEFTSHIFT-ALT-TAB
# 4  RIGHTSHIFT-ALT-TAB
# 5  CTRL-ESC
# 6  ALT-ESC
# 7  ALT-SPACE
# 8  (SHIFT)-CTRL-F1
# 9  (SHIFT)-CTRL-F2
# 10 (SHIFT)-CTRL-F3
# 11 (SHIFT)-CTRL-F4
```

**pcuhwsvc.ini (Vorlage im Verzeichnis ...\siemens\System\etc)**

```
# 12 (SHIFT)-CTRL-F5
# 13 (SHIFT)-CTRL-F6
# 14 (SHIFT)-CTRL-F7
# 15 (SHIFT)-CTRL-F8
# 16 M$_1
# 17 M$_2
# 18 CAPSLOCK
# 19 NUMLOCK
# 20 (reserved)

KeySequencesEnable = 0

SeqActive_0= 1
SeqActive_1= 1
SeqActive_2= 1
SeqActive_3= 1
SeqActive_4= 1
SeqActive_5= 1
SeqActive_6= 1
SeqActive_7= 1
SeqActive_8= 1
SeqActive_9= 1
SeqActive_11= 1
SeqActive_12= 1
SeqActive_13= 1
SeqActive_14= 1
SeqActive_15= 1
SeqActive_16= 1
SeqActive_17= 1
SeqActive_18= 1
SeqActive_19= 1
SeqActive_20= 1

# Autorepeat-Suppression of Function Keys F1-F12 and Hardkeys
#================================================
AutorepeatSuppression = 0
```

# 5.7 Installing STEP 7

## Requirement

- The Service Desktop is active
- The STEP 7 product DVD is available on the PC system, e.g. via an external DVD drive or network.

## Procedure

1. To start the installation assistant of STEP 7, on the product DVD in directory "`CD_1`", double-click `setup.exe`
2. Confirm the preselected directory under `C:\Program Files (x86)\Siemens\Step7\` as the installation directory.
3. In step "Transfer license keys", select "No, transfer license keys later".

   **Note**
   **Licensing of STEP 7 with SINUMERIK Add-on**

   The licensing for STEP 7 is performed during installation of the SINUMERIK Add-on.

4. After you have installed STEP 7, restart the PC system and switch to the Service Desktop.
5. To start the installation wizard of the SINUMERIK Add-on, double-click `setup.exe` on the product DVD in directory "`Sinumerik_Add_On`".
   The installation wizard guides you through the following steps.
6. After you have installed the SINUMERIK Add-on, restart the PC system.

## Result

STEP 7 has been installed with SINUMERIK Add-on.

The software has been registered in the Control Panel under the following designations:

- STEP 7 V5.5
- SINUMERIK Add-on for STEP7 V5.5

## 5.8 Linking the HMI software with STEP 7

If you want to make it possible to start STEP 7 directly from the user interface of the installed HMI software, add a softkey to it.

### Requirement

- STEP 7 V5.5 SP3 or higher is installed
- The SINUMERIK Add-on for STEP 7 V5.5 or higher is installed
- An HMI software version is installed (SINUMERIK Operate or HMI Pro sl)
- The Service Desktop is active

### Procedure

To add a softkey to the menu of the HMI software for starting STEP 7, proceed as follows:

1. Double-click the "`STEP7-Authorizing`" desktop icon.

### Result

A softkey for starting STEP 7 has been added to the HMI software. This softkey requires access level 3.

In `C:\Program Files (x86)\Siemens\MotionControl\compat\add_on\oemframe.ini`, the following settings have been made for this purpose:

```
[s7tgtopx]
; with HMI-Advanced: eliminate minimize- / maximize-buttons of the STEP7-window
WindowStyle_Off=196608
; with HMI-Advanced: switch to previous task when STEP7 is terminated
nSwitchToTaskAfterTermination= -2
```

**Note**

**Adjustments for OEM configurations**

The entries in `oemframe.ini` can be modified for OEM configurations.

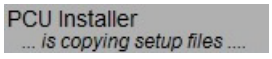
For more information about templates for parameterizing the system, see Section Directory structure and file conventions (Page 14).

# 5.9 PCU Installer

## 5.9.1 Overview

The PCU Installer assists you with the installation of setup packages on PC systems:

- The PCU Installer checks definable setup directories on the PC system, on removable disks, or in the network.
  These setup directories must not be write-protected.
- If setup packages exist, they are automatically installed one after the other. The order in which these packages are installed can be parameterized (Page 112).
- On the top right of the screen, a status display is displayed while the PCU Installer carries out such activities.




For further information on the activities of the PCU Installer, see Chapter Procedure example of an installation with the PCU Installer (Page 117).

**Overview of the configuration and operation**

- In the delivery condition of the PCU-Basesoftware, the PCU Installer is deactivated. You can activate the PCU Installer during the unattended installation of the PCU-Basesoftware via the response file (Page 20) or activate it manually after the attended installation (Page 108).
- You can make all settings in the PCU Installer in the associated configuration file (Page 112).
- Copy the setup packages to the defined directories to start the installation (Page 116).
- The PCU Installer consists of a client and a service, whereby setup packages can also be installed for a user account without administrator rights.

## 5.9.2 Activating the PCU Installer

### 5.9.2.1 Overview

In the delivery condition of the PCU base software, the PCU Installer is deactivated.

There are different ways of activating the PCU Installer:

- If you complete the installation of the PCU-Basesoftware unattended, you can have the PCU Installer activated via the `unattend.xml` response file. See Configuring an answer file (Page 20).
- If you complete the installation of the PCU base software with Windows 7 attended, or have not configured activation of the PCU Installer in the response file, you must activate the PCU Installer manually. To do this, use one of the following procedures:
  - Copy a preconfigured template of the PCUInst.ini to the "user" directory (Page 109)
  - Create your own version of the PCUInst.ini in one of the user directories and set the appropriate values
    See: Reference to the PCUInst.ini configuration file (Page 112)

#### 5.9.2.2 Activating via a configuration template

##### Requirement

- The PCU Installer is deactivated
- The Service Desktop is active
- The PCU Installer has not been configured yet and there is no adapted PCUInst.ini configuration file

##### Procedure

To activate the PCU Installer by copying a preconfigured template, proceed as follows:

1. Copy the following file to the clipboard: `D:\etc\Template_PCUInst.ini`
2. Paste the file into the following directory: `C:\ProgramData\Siemens\MotionControl\user\system\etc`
3. Rename the copied file to PCUInst.ini

##### Result

The PCU Installer has been activated and checks the setup directories during startup of the PC system, but not during system operation.

The copied configuration file only differs from the default configuration file by the following value:

```
[processing_loginphase]

StartState=activated
```

The settings of the copied configuration file are used because they are in a directory with higher priority than the default configuration file.

See: Directory structure and file conventions (Page 14)

To also use the PCU Installer during system operation, you must make an additional setting in the copied configuration file (Page 112):

```
[processing_systemphase]
StartState=activated
```

## 5.9.3 Deactivating the PCU Installer

In the delivery condition of the PCU-Basesoftware, the PCU Installer is deactivated.

**Note**

**Troubleshooting for incorrect installation routines**

If completion of an installation requires a restart, but the installation cannot be completed after the restart, the Service Desktop is not started.

In this case, restart the PC system in protected mode to prevent the PCU Installer from starting. (To start in protected mode, press F8 repeatedly during startup until a selection menu opens.)

Then reconfigure the PCU Installer or remove the setup packages.

### Requirement

- The Service Desktop is active
- Hidden directories are not shown (at least `C:\ProgramData\`)
  See also: Directories hidden in delivery condition (Page 14).
- A copy of the `PCUInst.ini` configuration file is stored in the `user` directory: `C:\ProgramData\Siemens\MotionControl\`**user**`\System\etc\PCUInst.ini`
- The `PCUInst.ini` configuration file in the `user` directory is not write-protected

### Procedure

To partly or completely deactivate the installation of the setup files by means of the PCU Installer, proceed as follows:

1. Open the `PCUInst.ini` configuration file in the `C:\ProgramData\Siemens\MotionControl\`**user**`\System\etc\PCUInst.ini` directory.
2. Deactivate the start of the PCU Installer in the desired phase:
   - To deactivate the PCU Installer during system operation, change the relevant key value in the `[processing_systemphase]` section: `StartState=stopped`.
   - To only deactivate the PCU Installer while a specific application is being executed (e.g. SINUMERIK Operate), enter the file name of the relevant application in the `[noactivation]` section, e.g. `APP001=run_hmi.exe`
     (You can specify any number of applications, e.g. `APP002=Siemens.Automation.Portal.exe`)
   - To deactivate the PCU Installer during run-up, change the relevant key value in the `[processing_loginphase]` section: `StartState=stopped`.
3. Save and close the file.

### Further information

For information on all configuration options, see Reference to the PCUInst.ini configuration file (Page 112).

## 5.9.4 Copying and adapting the configuration file

You can define PCU Installer operation in the associated configuration file. To do this, you must first copy the configuration file to a user directory.

Alternatively, you can copy a template in which the PCU Installer has already been activated (Page 109).

For example, you can configure the following in the configuration file:

- Which directories have been checked how many times for new setup packages.
- That installations are not started automatically while certain software is active.

A list of all setting options can be found under Reference to the PCUInst.ini configuration file (Page 112).

### Requirement

- The Service Desktop is active
- The PCU Installer has not been configured yet and there is no adapted PCUInst.ini configuration file

### Procedure

To copy and adapt the configuration file of the PCU Installer, proceed as follows:

1. Switch to the template directory: `C:\ProgramData\Siemens\MotionControl\siemens\System\etc\`
   The settings can be viewed in the template `PCUInst.ini`.
2. Save a copy of the template `PCUInst.ini` to one of the user directories:
   - `C:\ProgramData\Siemens\MotionControl\user\System\etc\`
   - `C:\ProgramData\Siemens\MotionControl\oem\System\etc\`
   - `C:\ProgramData\Siemens\MotionControl\addon\System\etc\`

**Note**

**Storage of the configuration file in the setup directory**

Alternatively, a configuration file can also be located directly with an installation setup.

In other words, if, for example, `E:\Setup` is a setup directory, this (or one of the direct subdirectories) can be the location of a configuration file. Settings made in this configuration file have a higher weighting than the settings in configuration files in the above-mentioned directories.

3. Adapt the values of the keys. You will find information about the syntax and setting options at Reference to the PCUInst.ini configuration file (Page 112) or in the comments of the configuration file.

   **Note**

   **The factory setting is taken if key pairs are missing**

   All key pairs that are not available in your file are taken automatically from the template in the `siemens` directory.

   Therefore, in the new file (e.g. `...\oem\System\etc\PCUInst.ini`), delete all the data except the section names and the keys that are to differ from the template.

   See also: Directory structure and file conventions (Page 14), Reference to the PCUInst.ini configuration file (Page 112)

4. Save and close the file.

### Result

The configuration file has been copied to a user directory and can be adapted.

You will find further information on the setting options in the following chapters.

## 5.9.5 Reference to the PCUInst.ini configuration file

You can define PCU Installer operation in the associated `PCUInst.ini` configuration file.

For example, you can configure the following:

- Which directories have been checked how many times for new setup packages.
- That installations are not started automatically while certain software is active.

The priority concept of this configuration file is the same as that of the other configuration files, see Directory structure and file conventions (Page 14).

You can also deactivate settings in `PCUInst.ini` by deleting the value of the key in the higher priority configuration file. For example, to deactivate `DIR002=C:\Setup` in a file of lower priority, enter `DIR002=` in the file with higher priority.

**Note**

**Maximum length of the file path**

The file path to a setup must not exceed 255 characters (including the file name).

Table 5-2 Setting options in the configuration file of the PCU Installer

| Section | Settings | Description |
|---|---|---|
| [local_setupdirs] | DIR001=<br>DIR002=<br>... | Specifies the local setup directories on the PC system to be checked.<br>The directories must not be write-protected.<br>You can specify several directories, which are then checked in the specified order.<br>You define the order with a number after "DIR".<br>Subdirectories at the first level are also scanned. |
| [removable_<br>setupdirs] | DIR001=<br>DIR002=<br>... | Specifies the setup directories on removable disks to be checked.<br>The directories must not be write-protected.<br>You can specify several directories, which are then checked in the specified order.<br>You define the order with a number after "DIR".<br>Subdirectories at the first level are also scanned.<br>If you do not know the drive letters to be checked, use the <REMOVABLE_MEDIA> drive variable, e.g.:<br>DIR001=<REMOVABLE_MEDIA>:\Install<br>The drive variable then represents all the removable disks that are known at the time in Windows. You can also specify fixed drive letters without the directories being checked twice.<br>The drive variable is only valid in the [removable_setupdirs] section. |
| [net_setupdirs] | DIR001=<br>DIR002=<br>... | Specifies the setup directories on network drives to be checked.<br>The directories must not be write-protected.<br>See also: Creating a shortcut to the network drive (Page 151)<br>You can specify several directories, which are then checked in the specified order.<br>You define the order with a number after "DIR".<br>Subdirectories at the first level are also scanned. |
| [setupdirs_<br>settings] | deleteTmpLocal<br>SetupDirs= | Temporary directories are created during installation of a setup from a removable disk, network drive or the default setup directory D:\Install<br>(See also: Procedure example of an installation with the PCU Installer (Page 117))<br>Here you can specify whether these temporary directories are to be deleted again after installation.<br>Possible values: yes; no |
| | keepDInstall<br>SubdirsWithPrefix= | You can specify that subdirectories of D:\Install with a particular prefix in the directory name are not to be deleted after setup is completed. |
| [setupnames] | NAME001=<br>NAME002=<br>... | Specifies which files are setup packages and are to be included in the check of setup directories.<br>You can specify entire file names with their file extensions or represent individual parts of them with wildcard character *. |
| [setupexecution] | checkSetupLogFile= | Specifies whether a check is to be performed as to whether the file has already been installed before the installation of a setup file.<br>The check is performed based on the computer name and the setup name of the setup-specific log file.<br>Possible values: yes; no |
| | copyToTmp<br>LocalSetupDir= | You can specify whether setup files on removable disks are to be copied to a temporary, local directory before installation.<br>Setup files on network drives are always copied before installation.<br>Possible values: yes; no |
| [noactivation] | APP001=<br>APP002= | Specifies which applications of the PCU Installer are inactive during operation.<br>This setting is required to prevent operation of the HMI software or another application from being interrupted during execution of a setup. |

| Section | Settings | Description |
|---|---|---|
| [processing_ loginphase] | StartState= | Specifies whether the PCU Installer is active before the Desktop runs up.<br>The setting StartState=activated is required, in particular, if an installation requires a cold start.<br>Possible values:<br>• activated PCU Installer starts setups<br>• stopped PCU Installer is stopped completely |
| | continueSetups= | Some setups can only be completed after the system has been restarted. In this case, the PCU Installer waits until you restart the PC system.<br>You can specify here whether such setups are to be continued after a restart.<br>(See also: Note: Troubleshooting for incorrect installation routines (Page 116))<br>Possible values: yes; no |
| | scanSetupDirs= | Specifies whether setup directories are to be checked and, if required, setup files executed before the Desktop run-up.<br>The following directories are checked if this setting has been activated:<br>• Local setup directories from the [local_setupdirs] setting are always checked.<br>• You can define separately whether additional directories on removable disks or in the network are to be checked. (See settings "scanRemovableDriveSetupDirs" and "scanNetworkDriveSetupDirs")<br>If you deactivate this setting, neither local directories, nor directories on removable disks or in the network are checked, i.e. the setting scanSetupDirs=no overwrites, for example, scanNetworkDriveSetupDirs=yes.<br>Possible values: yes; no |
| | scanRemovableDrive SetupDirs= | Specifies whether directories on removable disks are to be checked from the setting [removable_setupdirs] and, if required, setup files executed before the Desktop run-up.<br>The setting "scanSetupDirs" must also be activated so that these directories are checked.<br>Possible values: yes; no |
| | scanNetworkDrive SetupDirs= | Specifies whether network directories are to be checked from the setting [net_setupdirs] and, if required, setup files executed before the Desktop run-up.<br>The setting "scanSetupDirs" must also be activated so that these directories are checked.<br>Possible values: yes; no |
| | waitTimeDevicesDetection= | Specifies the maximum wait time for device detection (removable media) in the system run-up. This setting is only relevant if the PCU Installer searches for setups on removable media prior to the desktop run-up. (See settings "scanSetupDirs", "scanRemovableDriveSetupDirs" in the section [processing_loginphase] and section [removable_setupdirs])<br>Possible values:<br>• Positive integer (milliseconds)<br>• infinite |
| | waitTimeDriveLettersAssignment= | Specifies how long the PCU Installer will wait during the system run-up until the system has assigned a drive letter to all of the logical drives. This setting is only relevant if the PCU Installer searches for setups on removable media prior to the desktop run-up. (See settings "scanSetupDirs", "scanRemovableDriveSetupDirs" in the section [processing_loginphase] and section [removable_setupdirs])<br>Possible values:<br>• Positive integer (milliseconds)<br>• infinite |

| Section | Settings | Description |
|---|---|---|
| [processing_ systemphase] | StartState= | Specifies whether the PCU Installer is active during operation of the Desktop.<br>Possible values:<br>• activated PCU Installer starts setups<br>• stopped PCU Installer is stopped completely<br>You can define exceptions in Section "[noactivation]". |
| | scanSetupDirs= | Specifies whether setup directories are to be checked and, if required, setup files executed during Desktop operation.<br>The following directories are checked if this setting has been activated:<br>• Local setup directories from the [local_setupdirs] setting are always checked.<br>• You can define separately whether additional directories on removable disks or in the network are to be checked. (See settings "scanRemovableDriveSetupDirs" and "scanNetworkDriveSetupDirs")<br>If you deactivate this setting, neither local directories, nor directories on removable disks or in the network are checked, i.e. the setting scanSetupDirs=no overwrites, for example, scanNetworkDriveSetupDirs=yes.<br>Possible values: yes; no |
| | scanRemovableDrive SetupDirs= | Specifies whether directories on removable disks are to be checked from the setting [removable_setupdirs] and, if required, setup files executed during Desktop operation.<br>The setting "scanSetupDirs" must also be activated so that these directories are checked.<br>Possible values: yes; no |
| | scanNetworkDrive SetupDirs= | Specifies whether network directories are to be checked from the setting [net_setup-dirs] and, if required, setup files executed during Desktop operation.<br>The setting "scanSetupDirs" must also be activated so that these directories are checked.<br>Possible values: yes; no |
| | numberOfScanCycles | Specifies whether and how often the setup directories check for new setup packages.<br>Possible values:<br>• infinite Infinite<br>• Integer<br>You can define the waiting time between checks in key waitTimeBetweenScanCycles |
| | waitTimeBetween ScanCycles | Specifies how long the PCU Installer waits after the setup directories have been checked before checking them again.<br>Enter the waiting time in milliseconds.<br>In key numberOfScanCycles, you can specify how often the setup directories will be checked. |
| report | InfoLevel | Specifies how detailed the information is to be which is stored in the log files.<br>Possible values:<br>• 1: Only basic information on the actions of the PCU Installer is logged<br>• 2: All PCU Installer-specific information is logged<br>• 3: All PCU Installer-specific information and some trace information is logged<br>• 4: All PCU Installer-specific information and additional trace information is logged<br>• 5: All PCU Installer-specific information and all trace information is logged |

### 5.9.6 Example configuration: Installation behavior as for PCU base software Windows XP

If you want the automatic installation of setups to be configured in the same way as for the PCU base software Windows XP, you can use the following configuration example in your configuration file with higher priority.

```
[removable_setupdirs]
DIR001=E:\install
[processing_loginphase]
StartState=activated
scanRemovableDriveSetupDirs=yes
```

You can deactivate all other entries of a PCUInst.ini with lower priority by entering the appropriate key in your configuration file, but leaving the value empty. For example, to deactivate `DIR002=C:\Setup` in a configuration file of lower priority, enter `DIR002=` in your file.

You can make any number of adaptations in your file. See: Reference to the PCUInst.ini configuration file (Page 112)

### 5.9.7 Installing software with the PCU Installer

#### Requirement

- The PCU Installer is activated
  See: Activating the PCU Installer (Page 108)
- If you want to install multiple setup packages one after the other, you must have defined multiple setup directories in the configuration file
- The setup directories to be used are empty and not write-protected

#### Procedure

To make the PCU Installer accessible to setup packages, proceed as follows:

1. Copy the setup packages to be installed into the setup directories. Copy each setup package into a separate directory.

   **Note**

   **Maximum length of the file path**

   The file path to a setup must not exceed 255 characters (including the file name).

### Result

The installation of the setup packages is started.
After a setup has been completed or cancelled, a specific log file is stored in the associated setup directory. The general log files of the PCU Installer are also supplemented in the log directory.

#### Further information

- Log files (.log) (Page 118)
- Procedure example of an installation with the PCU Installer (Page 117)

---

**Note**

**Troubleshooting for incorrect installation routines**

If completion of an installation requires a restart, but the installation cannot be completed after the restart, the Service Desktop is not started.

In this case, restart the PCU 50.5 in protected mode to prevent the PCU Installer from starting. (To start in protected mode, press F8 repeatedly during startup until a selection menu opens.)

Then reconfigure the PCU Installer or remove the setup packages.

---

## 5.9.8 Procedure example of an installation with the PCU Installer

The following example describes the activities of the PCU Installer if you have stored a setup file on a removable disk or network drive after the activation and configuration of the PCU Installer.

In this procedure example, it is assumed that you have already performed the following steps:

- You have activated the PCU Installer
- You have activated the checking of the network directories in the settings of the PCU Installer and defined an additional setup directory on a network drive.
  In the example: F:\Setup with F: as network drive.
- You have stored a setup file in this additional setup directory (e.g. F:\Setup\Update.exe).
- The setup directory is not write-protected.

### Example: Activities of the PCU Installer during installation of a network drive

- The defined setup directories are checked.
- The Update.exe file is found in the F:\Setup directory.
  As the Update.exe file has not been installed yet, there is no setup-specific log file in the directory.
- The Update.exe file is copied to a local temporary directory, e.g. D:\Install39\Update.exe

- The copied setup file is executed from the local temporary directory, e.g. D:\Install39\Update.exe
- Log files are created after completion of the installation:
  - The general log files are extended
  - The setup-specific log file is created in the original setup directory F:\Setup and in the local temporary directory D:\Install39

  See: Log files (.log) (Page 118)

## 5.9.9 Log files (.log)

In order to be able to track which setup packages and updates have been installed at any time, the activities of the PCU Installer are recorded in various log files.

### Setup-specific log files

A separate setup-specific log file is created for each setup package that is executed by the PCU Installer. The name of the setup and the name of the computer are entered in the file name of the log.

| | |
|---|---|
| Archiving | The setup directory in which the associated setup file was originally located, and (if copied by the setup) the local temporary directory. |
| File name | The log file is designated according to the following scheme:<br><Name of the setup file>@<Computer name>.log, e.g. Update@MyP-CU50.log |

If you activate the checkSetupLogFile= setting in the "PCUInst.ini" configuration file (Page 112), then the setup-specific log file prevents the same setup being installed several times on the same computer. The check is performed based on the computer name and the setup name.

In the delivery condition of the base software, this setting is deactivated. The setting should remain deactivated for the series start-up of devices with the same name or for the repair of an installation.

### General log files

The general PCU Installer log files record the operation and the communication of the PCU Installer client and PCU Installer service. There are therefore separate files for the client and service.

| | |
|---|---|
| Archiving | `C:\ProgramData\Siemens\MotionControl\User\System\Log\` |
| File name | PCU Installer client: `PCUInstaller_C.log`<br><br>PCU Installer service: `PCUInstaller_S.log` |

# 5.10 Migration

## Basic procedure

To transfer data from a PCU with Windows XP to a Siemens PC system with Windows 7, store the data on a USB flash drive, for example, and then copy it to the PC system:

- For SINUMERIK Operate, use the functions for generating and reading in commissioning archives.
- For STEP 7 V5.x, copy the generated STEP 7 project data onto the PC system.

## Further information

- Information about how commissioning files work in **SINUMERIK Operate** can be found in the *SINUMERIK 840D sl SINUMERIK Operate Commissioning Manual (IM9)*, Chapter *Data backup*.
- Information on **STEP 7 V5.x** can be found in the associated online help.
- Information about other SINUMERIK applications is provided in the relevant documentation. You will find documentation on a large number of products in the SIEMENS Service&Support portal:
  Service&Support portal > CNC automation system SINUMERIK (http://support.automation.siemens.com/WW/view/en/10805517/133300)

# Backing up and restoring data

# 6

## 6.1 Overview

You can perform installation and commissioning work on the PC system in the Service Center, which is based on Microsoft Windows PE.

When the Desktop is active, configuration of the Service Center is possible via the "ServiceCenter Backup-Restore" (Page 62) icon.

Most of the functions of the Service Center are available in the main menu (Page 123).

In particular, you can create or restore disk images in the Service Center. The following options are available:

- Create a disk image of the SSD (Page 128)
- Restore a disk image of the SSD (Page 130)
- Create a disk image of a partition (Page 132)
- Restore a disk image of a partition (Page 134)

The Service Center uses the Symantec Ghost (Page 139) software, which you can also call separately.

## 6.2 Starting the Service Center

To start the Service Center, proceed in one of the following ways:

- To call up the Service Center on the SSD while the PC system is starting, select the "Booting Service System" during startup in the Windows boot menu.
- To call up the emergency boot system on a USB flash drive while the PC system is starting, set the service switch to E or press F12 during startup to show the boot selection.
- To switch to the Service Center while the Service Desktop (Windows 7) is active, click the "ServiceCenter Backup-Restore" icon on the Desktop and then click "Start ServiceSystem".

# 6.3 Functions of the Service Center

## Overview

In the upper section of the Service Center main menu, you can select one of five tasks. Buttons for additional functions are available in the lower section.

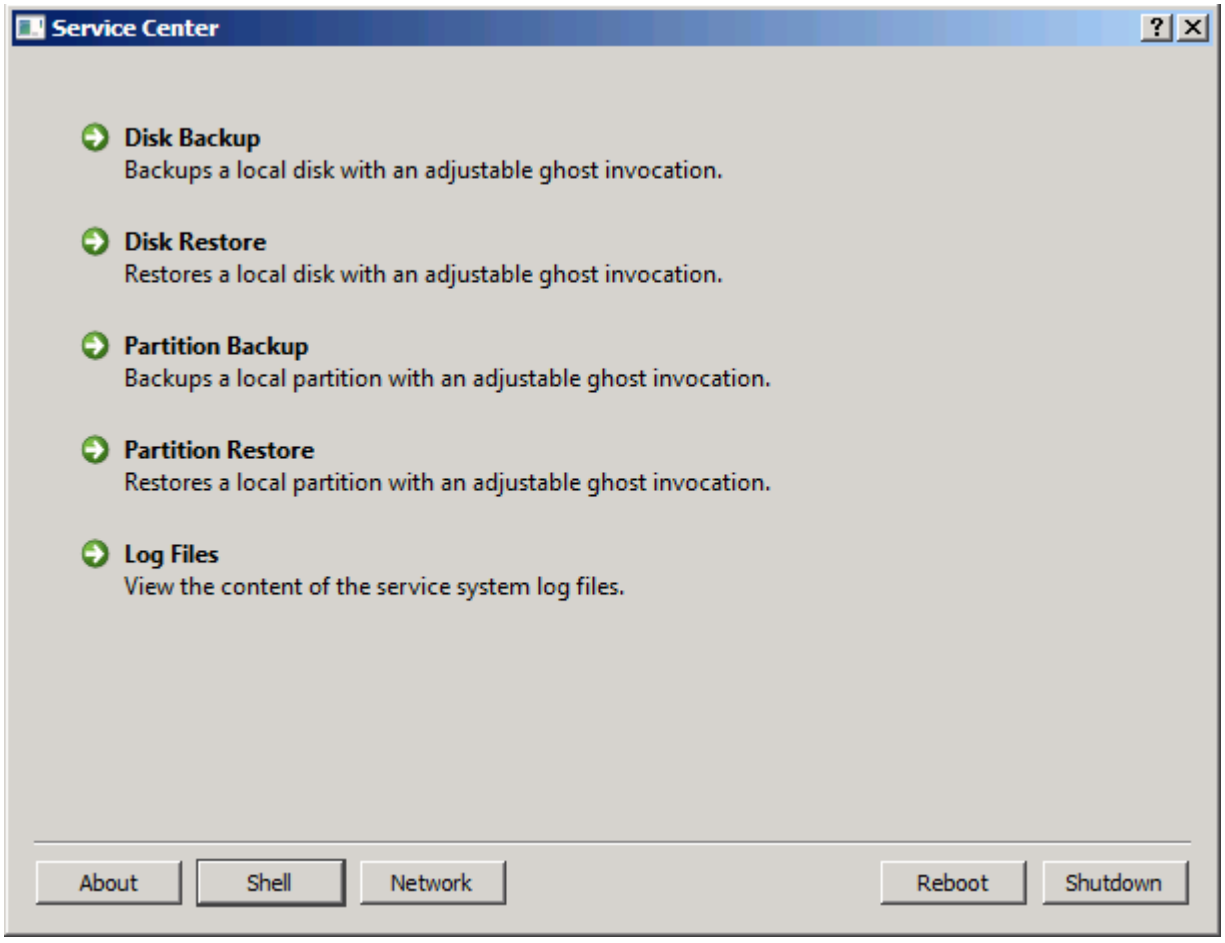



Figure 6-1 Service Center - Main menu

Table 6-1 Functions in the main menu of the Service Center

| Element of the window | Purpose |
|---|---|
| Disk Backup | Create a disk image of the local SSD.<br>With this command, you can create a disk image of the entire SSD (with all partitions). |
| Disk Restore | Restore the local SSD with an existing disk image.<br>With this command, you can restore the complete SSD (with all partitions) with a previously created disk image. |
| Partition Backup | Create a disk image of a partition on the local SSD.<br>With this command, you can create a disk image of an individual partition of the SSD. |
| Partition Restore | Restore a partition on the local SSD with a disk image.<br>With this command, you can restore an individual partition of the SSD with a previously created disk image. |

| Element of the window | Purpose |
|---|---|
| Log Files | View and save log files. |
| About | Viewing information about the Service Center. |
| Shell | Open the prompt.<br>With this command, you can enter commands for executing tasks on the PC system without using a graphical user interface. |
| Network | Making network settings (Page 125).<br>Any network settings you make in this dialog box are temporary. To change network settings permanently, switch to the Service Desktop. |
| Reboot | Restart the PC system. |
| Shutdown | Shut down the PC system. |

**Note**

**Data backup and restore with CompactFlash cards**

If there is a CompactFlash card in the slot of the PC system, you will not be able to back up or restore data via the Service Center.

Instead, start the "ghost32.exe" program via the prompt of the Service Center.

# 6.4 Network settings in the Service Center

You can make network settings in the main menu of the Service Center by clicking "Network":

- IP settings at the enterprise network interface X1 (Local Area Connection 2)
- IP settings at the plant network interface X2 (Local Area Connection)
- Configuration of the host
- Configuration of IP routing

**Note**

**Temporarily storing network settings in the Service Center**

Any network settings you make in this dialog box are immediately effective, but temporary.

To change network settings permanently, switch to the Service Desktop.

For further information, see Section Configuration of the Service Center (Page 62).

## Overview

### IP settings

You can make the following IP settings in the "Network Settings" dialog box:

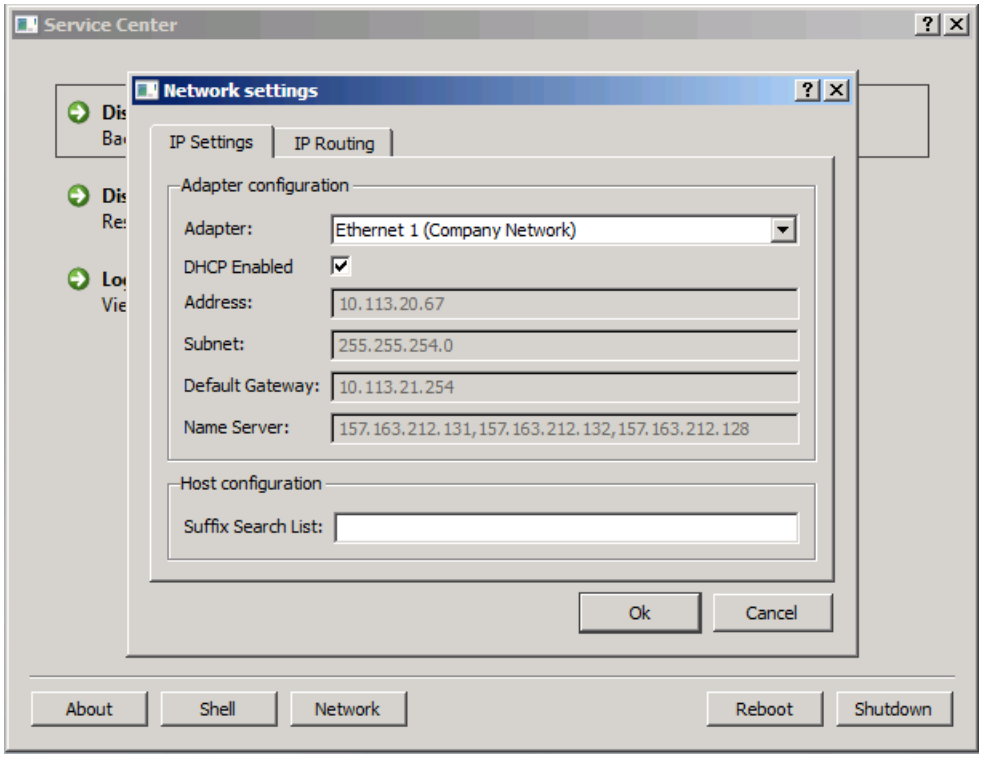



Figure 6-2 Service Center - IP settings

Table 6-2 IP settings in the "Network Settings" dialog box

| Section | Settings | Purpose |
|---|---|---|
| Adapter configuration | Adapter | Select the Ethernet interface that you want to configure.<br>All settings in Section "Adapter configuration" refer to the selected Ethernet interface. |
| | DHCP Enabled | Select whether the addresses are to be fetched dynamically from the DHCP server.<br>If the checkbox is deactivated, you must make the settings manually. |
| | Address | Set the IP address of the NCU.<br>You can specify an IP address from the following range:<br>192.168.214.250 - 254 |
| | Subnet | Specify a subnet mask, e.g. 255.255.255.0 |
| | Default Gateway | Set the IP address of the standard gateway.<br>A standard gateway creates a standard route in the IP routing table for all destinations that are not in the subnet. |
| | Name Server | Set the IP address of the name server.<br>A name server answers questions asked about a domain name zone using a DNS database. |
| Host configuration | Suffix Search List | Parameterize a search list for DNS suffix, e.g. "network.com".<br>The factory setting of the Ethernet interface "Local Area Connection" is ".local". |

### Routing settings

You can make the following routing settings in the "Network Settings" dialog box:

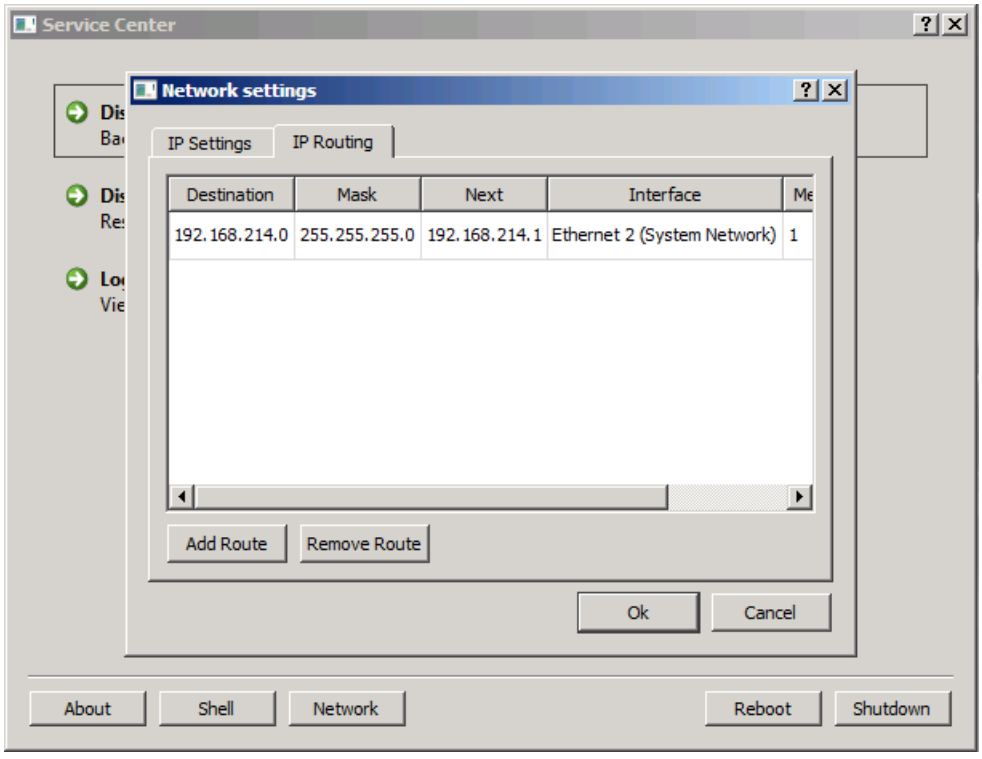



Figure 6-3 Service Center - IP routing

Table 6-3 Routing settings in the "Network Settings" dialog box

| Settings | Purpose |
|---|---|
| Add Route | Add a new route. |
| Remove Route | Delete a selected route. |
| Destination | Specify the network destination of the route as an IP address. |
| Mask | Subnet mask of the specified IP addresses. |
| Next | Next hop or forwarding IP address via which the network destination can be accessed. |
| Interface | Select an interface to which the configured IP routing applies:<br>• Enterprise network interface X1 (Local Area Connection 2)<br>• Plant network interface X2 (Local Area Connection) |
| Metric | Integer cost metric (1 to 9999). Required if several routes for sending a packet to a network destination are possible in the routing table. |

## See also

Network settings (Page 10)
Release directory of the PC/PC in the network (Page 148)
Creating a shortcut to the network drive (Page 151)
Configuring routing in the network (Page 144)

## 6.5 Create a disk image of the SSD

You can create a disk image of the entire SSD in the Service Center with the "Disk Backup" function.

### Procedure

To save the SSD as a disk image, proceed as follows:

1. Start the Service Center, e.g. via the Windows boot menu.
   See also: Starting the Service Center (Page 122)
2. In the main menu of the Service Center, click "Disk Backup".
   The "Service Center - Disk Backup" dialog box opens.

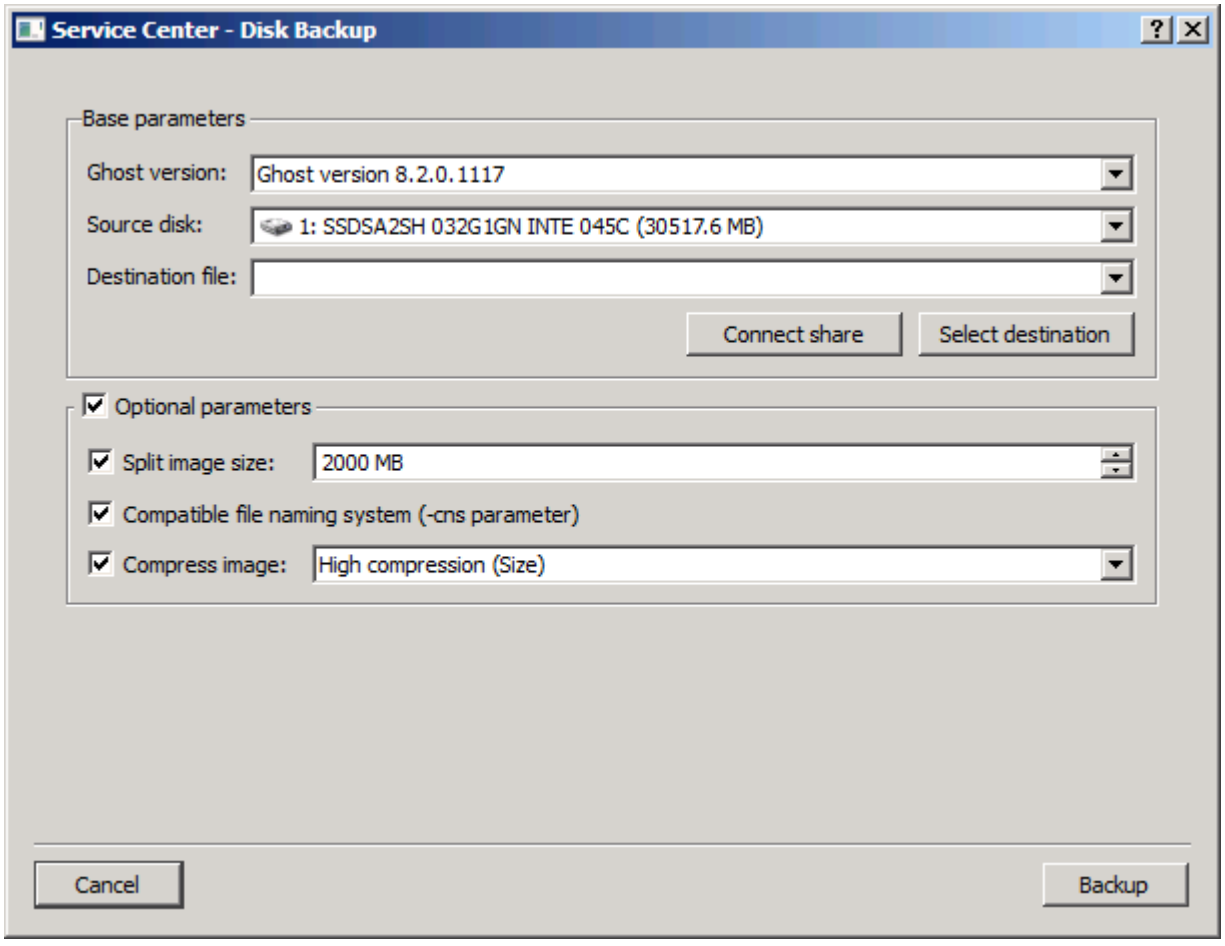



Figure 6-4 Service Center - Disk Backup

3. Make the required settings here:

| Section | Settings | Purpose |
|---|---|---|
| Base parameters | Ghost version | Select the Symantec Ghost version that is to be used when creating the disk image:<br>• Version for Windows 7: "Ghost version 11.5.x"<br>• Version for Windows XP: "Ghost version 8.2.x" |
| | Source disk | Select a data medium of which you want to create a data image. |
| | Destination file | Enter a destination directory and file name.<br>Disk images can be stored either locally or in the network. |
| Base parameters > Destination file | Connect share | Open the "Connect a Share" dialog box (Page 137).<br>You can set up a shared resource (e.g. a directory in the network), to which you can store the disk image. |
| | Select destination | Open the "Select Destination File" dialog box.<br>You can navigate through a local directory of the PC system and enter a file name. |
| Optional parameters | Split image size | Split the disk image and select a split image size.<br>You can split the disk image into several files of a defined size.<br>Split the disk image size in the following cases:<br>• If the file system (e.g. FAT32) can only manage files of a certain size.<br>• If the disk image is to be archived onto multiple data media, for example, when using CDs. |
| | Compatible file naming system | Select the compatibility of the name scheme.<br>This option is required so that a disk image with older versions of the "PCU-Basesoftware Windows XP" can be read in by Windows XP.<br>It is not possible to use a different operating system on a Siemens PC system other than the preinstalled system (e.g. Windows XP on a PCU 50.5 Windows 7). |
| | Compress image | Activate Compression or select a compression level.<br>Higher compression results in a smaller file size, but increases the time needed to compress or decompress the data. |

4. Confirm your settings with "Backup" to start generation of the disk image.

# 6.6 Restore a disk image of the SSD

If your system is no longer functioning stably, you can restore it with a disk image.

**Note**

**Restoring replaces all current files**

When you restore the PC system from a disk image, all programs, settings and files are completely replaced by the disk image. You cannot restore individual files or exclude individual files from restoration.

## Requirement

A disk image exists
See: Create a disk image of the SSD (Page 128)

## Procedure

To restore a disk image, proceed as follows:

1. Start the Service Center, e.g. via the Windows boot menu.
   See also: Starting the Service Center (Page 122)
2. In the main menu of the Service Center, click "Disk Restore".
   The "Disk Restore" dialog box opens.

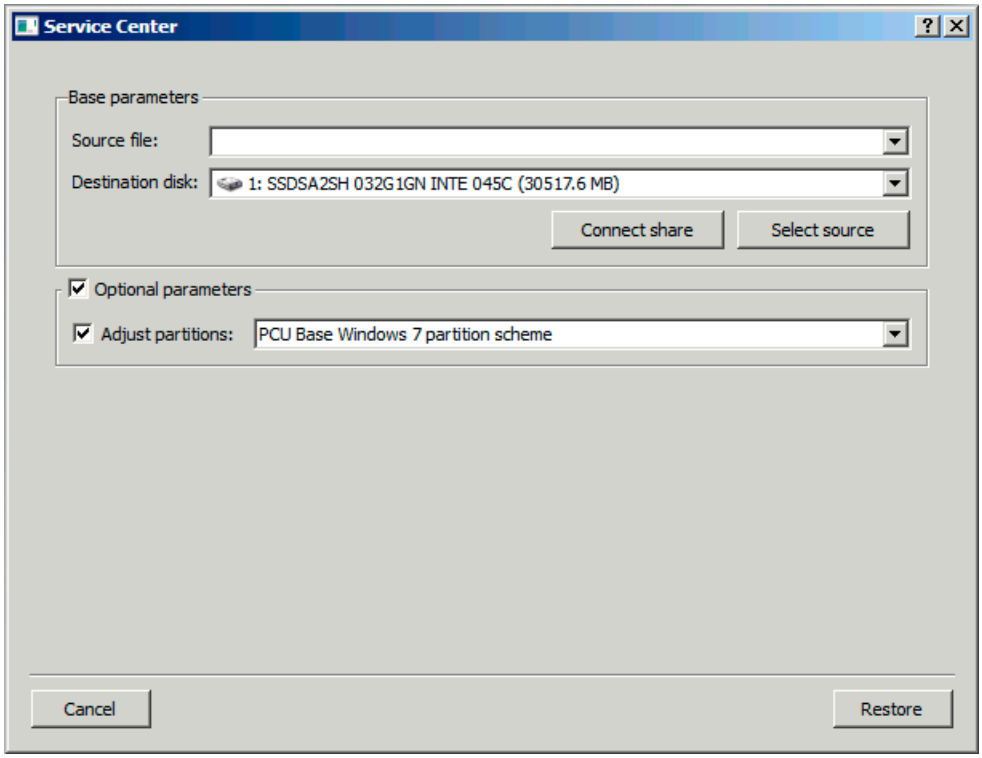



Figure 6-5 Service Center - Disk Restore

3. Make the required settings here:

| Section | Settings | Purpose |
|---|---|---|
| Base parameters | Source file | Select the disk image from which the SSD is to be restored.<br>You can use local disk images or disk images stored in the network. |
| | Destination disk | Select the SSD to be restored from the disk image. |
| Base parameters > Destination disk | Connect share | Open the "Connect a Share" dialog box (Page 137).<br>You can manage a shared resource (e.g. a directory in the network), to use a disk image stored on the network to restore the SSD. |
| | Select source | Open the "Select Source File" dialog box.<br>You can navigate through a local directory of the PC system and select a disk image for restoration. |
| Optional parameters | Adjust partitions | Select the partition scheme to be used when restoring the disk image on a PCU 50.x:<br>• If the PCU-Basesoftware Windows 7 is on the disk image, choose "PCU Base Windows 7 partition scheme".<br>• If the PCU-Basesoftware Windows XP is on the disk image, choose "PCU Base Windows XP partition scheme".<br>It is not possible to use a different operating system on a Siemens PC system other than the preinstalled system (e.g. Windows XP on a PCU 50.5 Windows 7).<br>If you are using a different PC system than the PCU 50.5, you must deactivate the "Adjust partitions" setting. |

4. To start the restore process and entirely replace the existing data on the SSD with the disk image, confirm the settings with "Restore".

## Result

The restore process is started.

**Note**

**Troubleshooting if reading in of the disk image fails**

If data transfer is interrupted during the restore process, the PC system will have no executable system.

In this case, use a bootable USB flash drive with Service Center to read the disk image in again.

## 6.7 Create a disk image of a partition

You can create a disk image of an individual partition of the SSD in the Service Center with the "Partition Backup" function.

### Procedure

To save a partition of the SSD as a disk image, proceed as follows:

1. Start the Service Center, e.g. via the Windows boot menu.
   See also: Starting the Service Center (Page 122)
2. In the main menu of the Service Center, click "Partition Backup".
   The "Service Center - Partition Backup" dialog box opens.

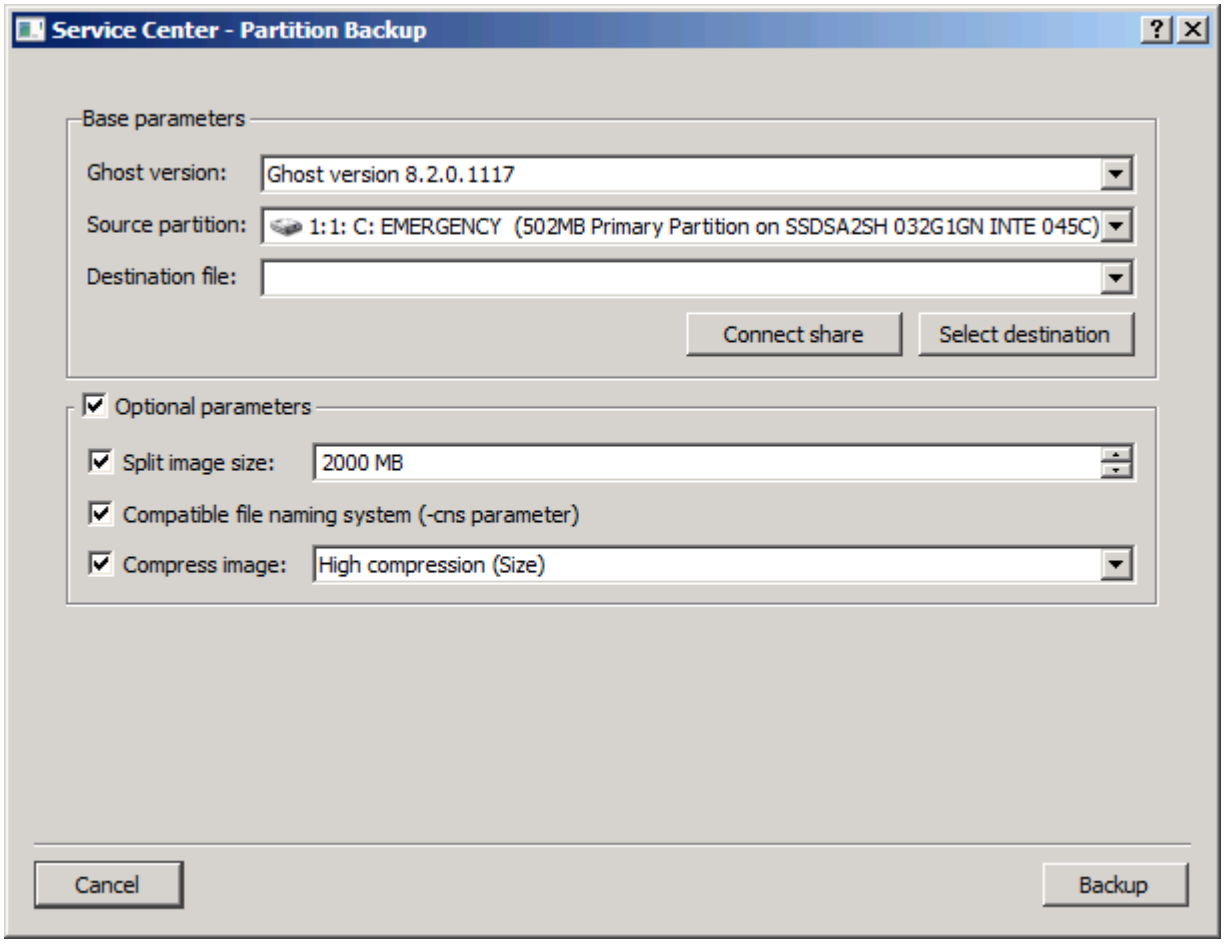



Figure 6-6 Service Center - Partition Backup

3. Make the required settings here:

| Section | Settings | Purpose |
|---|---|---|
| Base parameters | Ghost version | Select the Symantec Ghost version that is to be used when creating the disk image:<br>• Version for Windows 7: "Ghost version 11.5.x"<br>• Version for Windows XP: "Ghost version 8.2.x" |
| | Source partition | Select a data medium of which you want to create a data image. |
| | Destination file | Enter a destination directory and file name.<br>Disk images can be stored either locally or in the network. |
| Base parameters > Destination file | Connect share | Open the "Connect a Share" dialog box (Page 137).<br>You can set up a shared resource (e.g. a directory in the network), to which you can store the disk image. |
| | Select destination | Open the "Select destination file" dialog box.<br>You can navigate through a local directory of the PC system and enter a file name. |
| Optional parameters | Split image size | Split the disk image and select a split image size (PCU 50.5 only).<br>If you are using a PCU 50.5, you can split the disk image into several files of a defined size.<br>Split the disk image size in the following cases:<br>• If the file system (e.g. FAT32) can only manage files of a certain size.<br>• If the disk image is to be archived onto multiple data media, for example, when using CDs.<br>If you are using a different PC system than the PCU 50.5, you must deactivate the "Split image size" setting. |
| | Compatible file naming system | Select the compatibility of the name scheme.<br>This option is required so that a disk image with older versions of the "PCU-Basesoftware Windows XP" can be read in by Windows XP.<br>It is not possible to use a different operating system on a Siemens PC system other than the preinstalled system (e.g. Windows XP on a PCU 50.5 Windows 7). |
| | Compress image | Activate Compression or select a compression level.<br>Higher compression results in a smaller file size, but increases the time needed to compress or decompress the data. |

4. Confirm your settings with "Backup" to start generation of the disk image.

## 6.8 Restore a disk image of a partition

If your system is no longer functioning stably, you can restore it with a disk image.

**Note**

**Restoring replaces all current files**

When you restore a partition from a disk image, all programs, settings, and files are completely replaced on this partition by the disk image. You cannot restore individual files or exclude individual files from restoration.

### Requirement

A disk image of a partition exists
See: Starting the Service Center (Page 122)

## Procedure

To restore a partition from a disk image, proceed as follows:

1. Start the Service Center, e.g. via the Windows boot menu.
   See also: Create a disk image of a partition (Page 132)
2. In the main menu of the Service Center, click "Partition Restore".
   The "Partition Restore" dialog box opens.

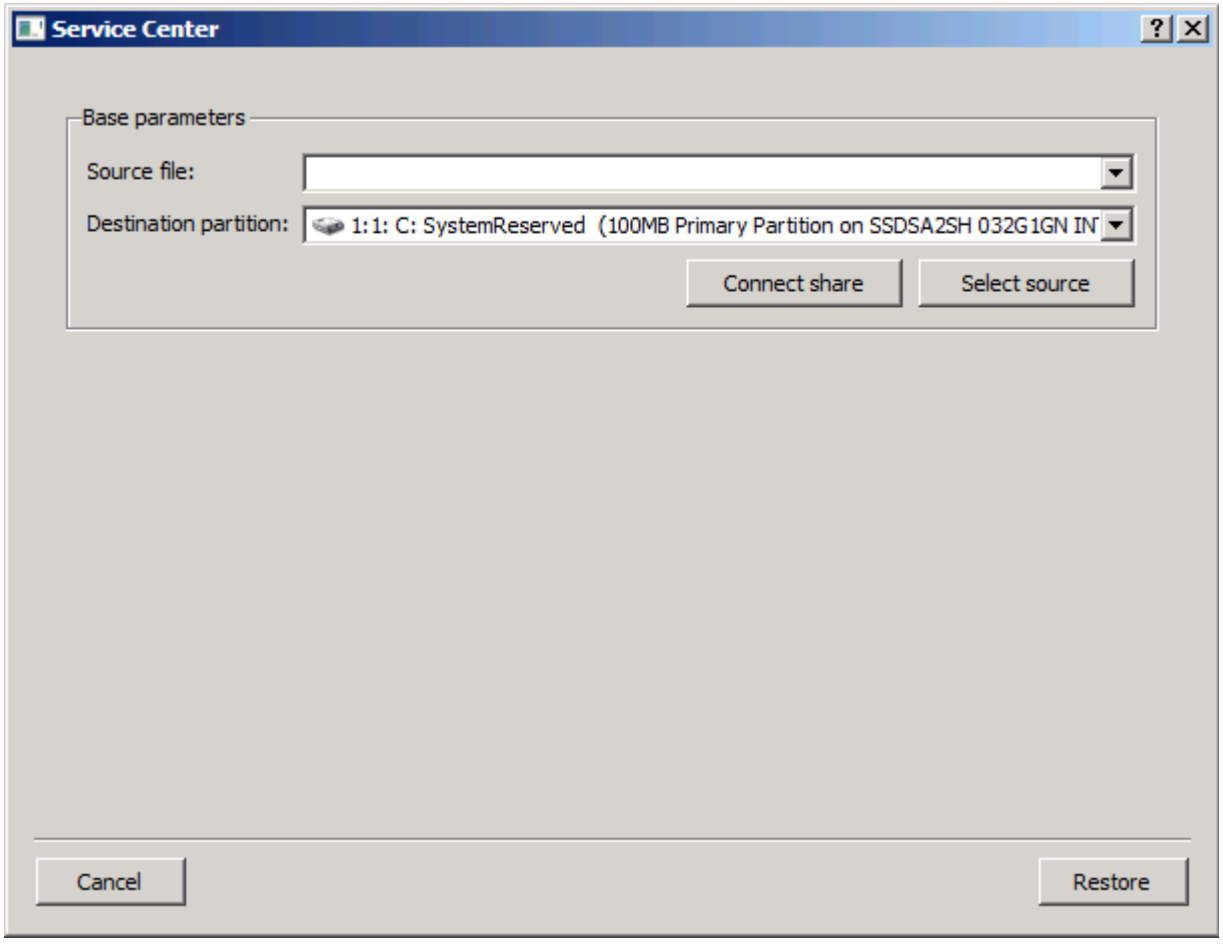



Figure 6-7 Service Center - Disk Restore

3. Make the required settings here:

| Section | Settings | Purpose |
|---|---|---|
| Base parameters | Source file | Select the disk image from which the partition is to be restored.<br>You can use local disk images or disk images stored in the network. |
| | Destination partition | Select the partition to be restored from the disk image. |
| Base parameters > Destination partition | Connect share | Open the "Connect a Share" dialog box (Page 137).<br>You can manage a shared resource (e.g. a directory in the network), to use a disk image stored on the network to restore the SSD. |
| | Select source | Open the "Select source file" dialog box.<br>You can navigate through a local directory on the PC system and select a disk image for restoration. |

4. To start the restore process and entirely replace the existing data on the partition with the disk image, confirm the settings with "Restore".

## Result

The restore process is started.

**Note**

**Troubleshooting if reading in of the disk image fails**

If data transfer is interrupted during the restore process, the PC system will have no executable system.

In this case, use a bootable USB flash drive with Service Center to read the disk image in again.

# 6.9 Network settings in the Service Center

To manage a disk image in the network, you can display a shared network directory on the PC system as a drive.

In the Service Center, you configure network drives in the "Connect a share" dialog box. You can call this dialog box directly from the relevant dialog boxes to both create (Page 128) and restore (Page 130) disk images.

## Requirement

The network directory is shared with the relevant user accounts in the domain and on the PC system.
See also: Release directory of the PC/PC in the network (Page 148)

## Overview

The setting options in the "Connect a share" dialog box have the following meaning:

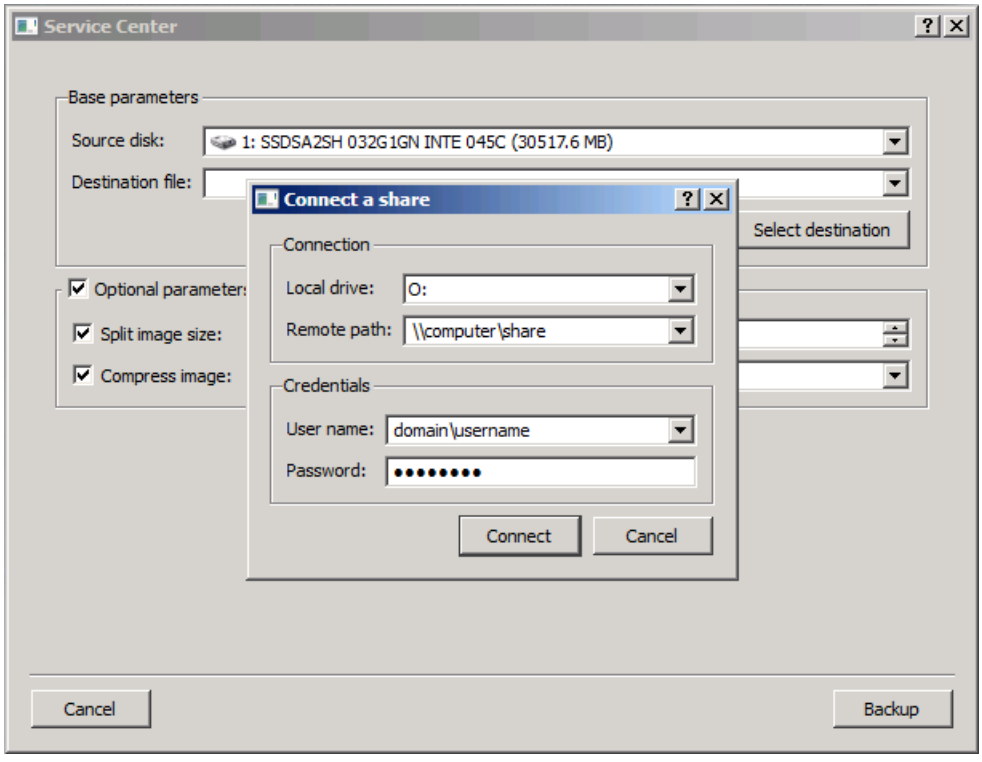



Figure 6-8 "Connect a Share" dialog box

Table 6-4 Settings in the "Connect a Share" dialog box in the Service Center

| Section | Settings | Purpose |
|---|---|---|
| Connection | Local drive | Select drive letters under which the network directories are to be accessible on the PC system.<br><br>The drive letters Z, Y and X are assigned to USB interfaces on SINUMERIK TCUs and operator panels. Do not use these drive letters for other purposes in order to avoid problems when using these USB interfaces. |
| | Remote path | Specify the device in the network and share name of the resource.<br>Syntax: `\\Name of the server\share name of the network directory`<br>Example: `\\Backup_Server\PCU_Backup` |
| Credentials | User name | Enter the user account under which the resource in the network can be accessed.<br>If you have shared the resource on the server for certain users only, you must enter one of these user accounts here. |
| | Password | Enter the password of the user account. |

## Further information

You will find information about releasing directories in the network and network drives in the following chapters:

- Release directory of the PC/PC in the network (Page 148)
- Creating a shortcut to the network drive (Page 151)

## 6.10 Starting Symantec Ghost directly

With the "Symantec Ghost" software you can store the entire contents of an SSD as a disk image. Disk images can be kept on various storage media for subsequent restoration of the data on the SSD.

Symantec Ghost is included in the PCU base software scope of delivery and on the spare part SSD for PCU. Symantec Ghost is used by the components of the PCU base software to back up or restore data.

You can also start Symantec Ghost directly from the prompt to execute the data backup tasks from the user interface of Ghost. This is necessary, for example, when using CompactFlash cards.

### Procedure

To start Symantec Ghost directly, proceed as follows:

1. Call the prompt from the Service Center or the Service Desktop.
2. Enter "ghost.32.exe" and confirm the entry.

### Further information

More information is available in the Internet on website Ghost.com (http://www.ghost.com).

# 6.11 Bootable USB flash drive

## Usage

You can generate a bootable Service Center (Emergency Boot System) based on Microsoft Windows PE on a USB flash drive as preventive measure against a defect.

The Ghost image for the creation of the bootable USB flash drive is on the installed NCU-Basesoftware at: `D:\Eboot\eboot.gho`

## Requirement

- The Windows 7 Service Desktop is active
- You have a USB flash drive (e.g. SIMATIC PC USB flash drive)

## Procedure

To create a bootable USD flash drive, proceed as follows:

1. Connect the USB flash drive to a USB interface of the PC system.
2. Start Symantec Ghost, for example, by entering `Ghost32.exe` in the search bar in the start menu.
3. In Symantec Ghost, select the "From Image" command from the "Local > Disk" menu.
4. Select `D:\Eboot\eboot.gho` as the source and the USB flash drive as the destination.

## Result

The Emergency Boot System is installed on the USB flash drive and the data medium is renamed to "EBOOT". The Service Center on the bootable USB flash drive can now be used.

# 6.12 Operating a service PC/PC in the network

## 6.12.1 Overview

### System network

In the following applications, you will need, for example, a connection between the PC system and a PG/PC in the system network:

- Save the disk image of the SSD to a PG/PC
- Restore the SSD of a PC system via the DVD drive of a PG/PC
- Start up the replacement SSD
- Install the software on the PC system via the DVD drive of a PG/PC

### Company network

If you want to connect the service PG/PC via a company network (Local Area Connection 2; interface X1), contact your network service center.

More information on the Windows Firewall settings in the company network can be found in the section Adapting the firewall settings (Page 66).

## 6.12.2 Connection options in the system network

### Overview

The following figures show the typical connection options in the system network:

- Service PG/PC with (twisted) Ethernet cable direct to interface "X2" of the Siemens PC system.
- Service PG/PC with (non-twisted) Ethernet cable via switch to interface "X2" of the Siemens PC system.

### Configuration with PG/PC direct to the Siemens PC system (e.g. PCU 50.5)

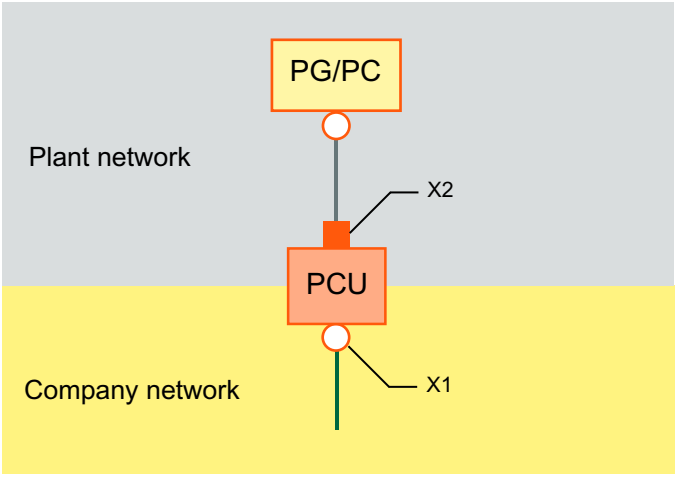



Figure 6-9 Connecting a PG/PC directly to PCU 50.5

### Configuration with PG/PC and switch direct to the Siemens PC system (e.g. PCU 50.5)

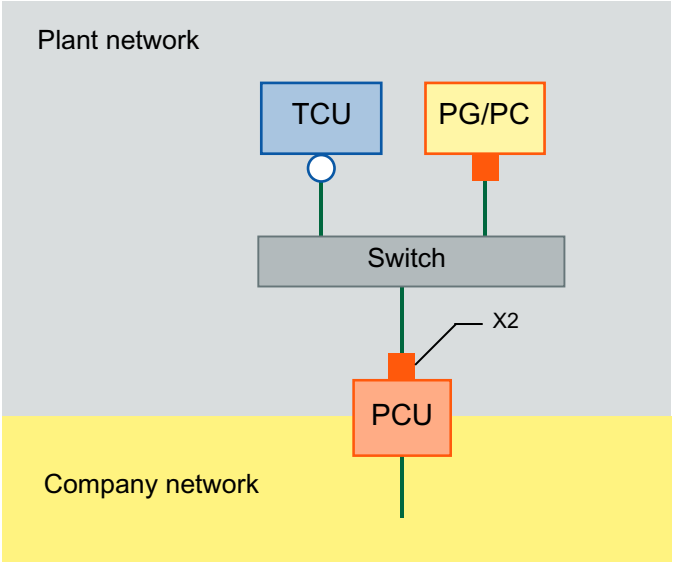



Figure 6-10 Connecting a PG/PC to PCU 50.5 via a switch

**Note**

If a PCU without its own OP/TP including TCU is switched off and then back on, and if this is to be followed by the power up of the PCU of the service system (EBS), an external screen and keyboard are needed to operate the PCU.

An external screen and keyboard are not needed if the TCUs are operating and the PCU is not switched off with the EBS during booting.

## Basic procedure

On the PG/PC with Windows 7:

- Connecting a PG/PC to a Siemens PC system according to one of the configurations shown above.
- The network protocol used is: TCP/IP.
  TCP/IP is already pre-configured on the Siemens PC system.
- Setting up IP addresses on the same subnetwork.
- Releasing a directory on the PG/PC for network access.

On the Siemens PC system in the Service Center (WinPE):

- On the PCU start the Service Center (WinPE).
- Establish a network connection to the shared directory of the PG/PC.
  See: Functions of the Service Center (Page 123)

## Further information

- Further information on the permissible network structures and network configurations can be found in the SINUMERIK 840D sl Operator Components and Networking Manual.

## 6.12.3 Configuring routing in the network

### Overview

If you have connected a PG/PC at interface X127 of the NCU, configure the network routing to establish the connection between the PG/PC and the PCU.

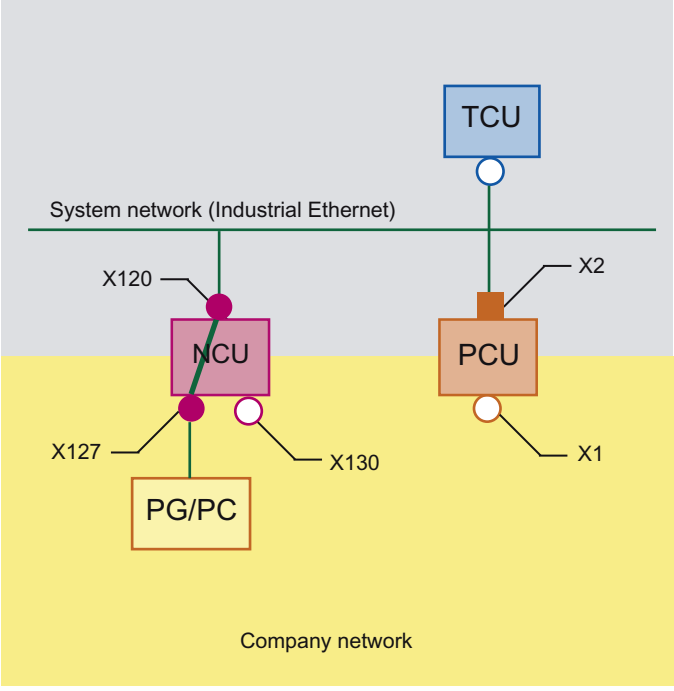



Figure 6-11 A networking example using IP routing

### Conditions

The following conditions must be met in order to use this function:

- The PG must be connected to interface X127 of an NCU in the system network.
- The NCU and PCU must be connected via the system network.
- On the NCU, the routing is activated in one of the following ways:
  - In SINUMERIK Operate 4.5 SP2 or higher, the routing is configured in image "Settings of the system network".
  - In the configuration file basesys.ini, key EnableSysNetToIBNForwarding=1 is set.
- If you wish to access a network directory, it must be shared.
- The user account in question is known to the PG/PC.
- The Service Center is active.

### Procedure

To configure the IP routing in the Service Center, proceed as follows:

1. In the main menu of the Service Center, click "Network".
2. In the "Network Settings" dialog box, click the "IP Routing" tab.

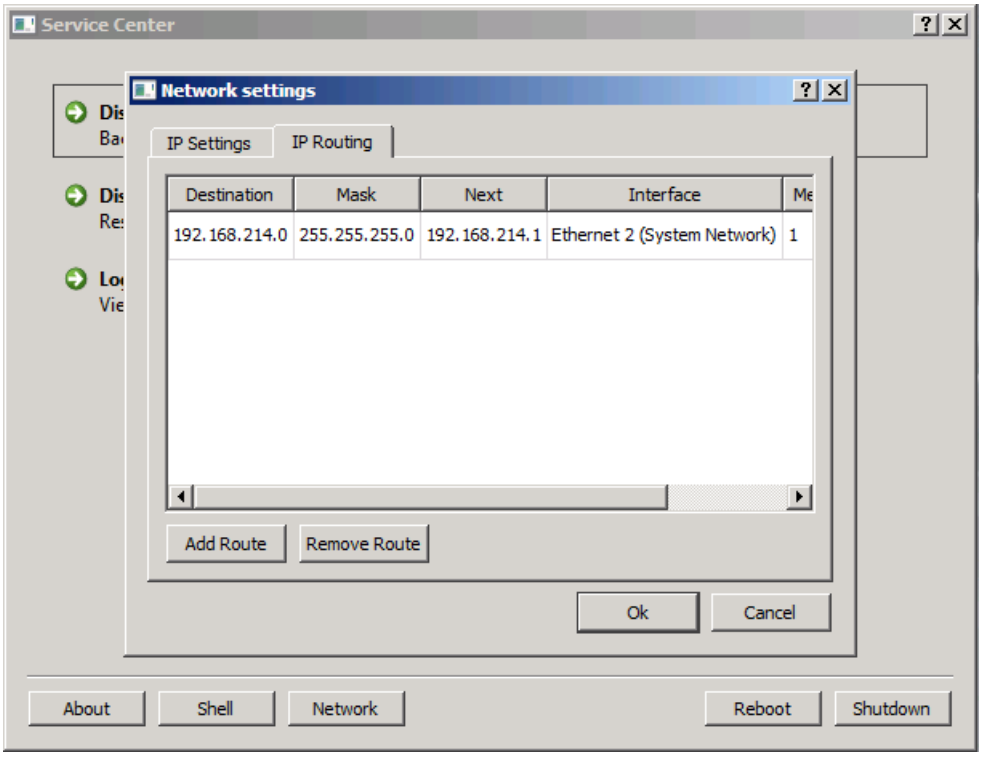



Figure 6-12 Service Center - IP routing

3. Configure the necessary routes. See also: Network settings in the Service Center (Page 125)

### Result

The routing has been configured and the PG/PC and Siemens PC system are connected via the NCU.

Now, perform the desired steps, for example, a Restore a disk image of the SSD (Page 130).

## 6.12.4 Configure the network settings of the PG/PC

Once you have connected a PG/PC in the network, make the network settings on the PG/PC.

The basic procedure for setup under Windows 7 or Windows XP is described below. Detailed information from Microsoft is provided in Windows under "Help and Support" or in the Internet: Microsoft Windows support (http://windows.microsoft.com/en-US/windows/windows-help#windows=windows-7)

## Precondition

- The PG/PC is linked with the Siemens PC system in one of the following ways:
  - The PG/PC is connected to the PC system with an Ethernet cable (either directly or via a switch)
    See: Connection options in the system network (Page 141)
  - The PG/PC is networked with the PC system via an interface on the NCU and the routing is configured.
    See: Configuring routing in the network (Page 144)
- A current Microsoft Windows operating system is installed on the PG/PC
- The PG/PC is switched on

## Procedure

To configure the network settings of the PG/PC for connecting a PC system in the system network, proceed as follows:

1. Open the "network connections" window:
   - To do this under Windows XP: in the "Start > Control Panel > Network and Internet Connections" menu, choose "Network Connections."
   - To do this under Windows 7: in the Control Panel in category "Network and Internet > View network status and tasks," choose "Change adapter settings."
2. Right-click on the network adapter you want to configure, then choose "Properties" from the shortcut menu.
   This opens the "Properties of <connection>" dialog box.

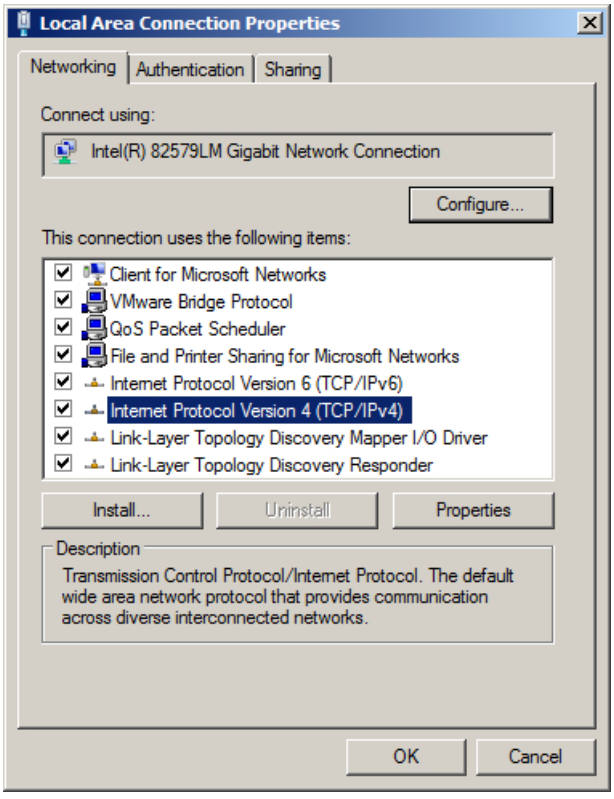



Figure 6-13 Properties of <connection>

3. Make the following settings:
   - Click "File and Printer Sharing for Microsoft Networks."

– To change the IP address of the PG/PC, double-click "Internet Protocol Version 4 (TCP/IPv4)", then configure the settings as follows:

| Section | Element | Settings |
|---|---|---|
| Use the following IP address | IP address | The following IP addresses in the system network can be used for the PG/PC:<br>• 192.168.214.250<br>• 192.168.214.251<br>• 192.168.214.252<br>• 192.168.214.253<br>• 192.168.214.254 |
| | Subnet mask | • 255.255.255.0 |

### Result

The network settings of the PG/PC are now configured for use with the Siemens PC system.

## 6.12.5 Release directory of the PC/PC in the network

If you want to, for example, store disk images on a PG/PC that you have linked to the Siemens PC system via the network, set up a network drive.

As soon as it has been set up, a new drive will appear on the PC system, which is really a directory on the PG/PC.

The basic procedure for setup under Windows 7 and Windows XP is described below. Detailed information from Microsoft is provided in Windows under "Help and Support" or in the Internet: Microsoft Windows support (http://windows.microsoft.com/en-US/windows/windows-help#windows=windows-7)

### Requirement

- The PG/PC is switched on.
- There is sufficient space on the PG/PC for saving, for example, disk images from the PCU.
- The user accounts to which you want to grant share rights are known in the domain or on the Siemens PC system.

### Procedure

Proceed as follows to share a directory for use in the network:

1. On the PG/PC, in the Control Panel call up Computer Management.
2. Under "System > Shared Folders" select the file "Shares."
3. In the "Action" menu, click the "New File Share..." command.
   The "Create A Shared Folder Wizard" opens.

4. Under step "Name, Description, and Settings," make a note of the Share path, which is a combination of the computer name and the share name. You will have to specify this name when you create a link to the network drive on the Siemens PC system.

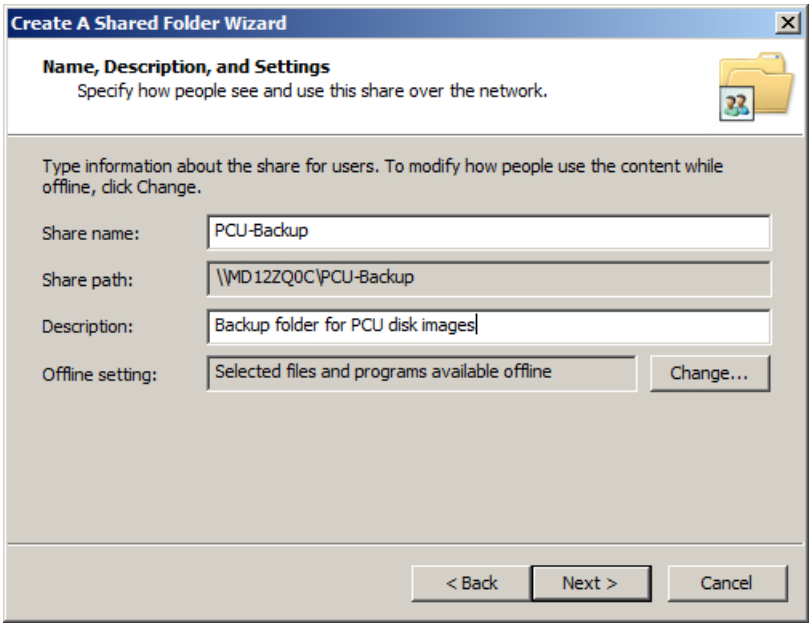



Figure 6-14 Create A Shared Folder Wizard - Name, Description, and Settings

5. Under step "Shared Folder Authorizations," select the "Customize authorizations" radio button, then click "Custom..."

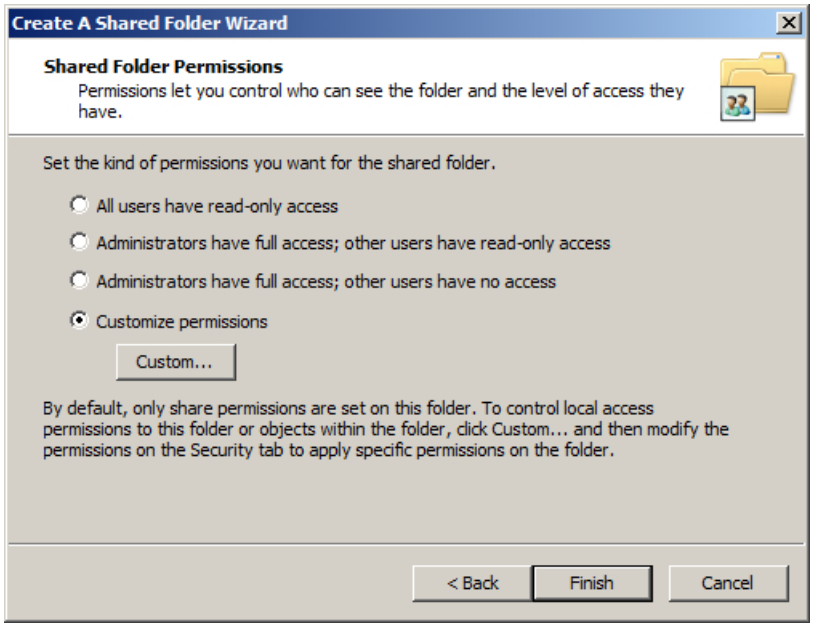



Figure 6-15 Create a Shared Folder Wizard - Shared Folder Authorizations

The "Customize Authorizations" dialog box opens.

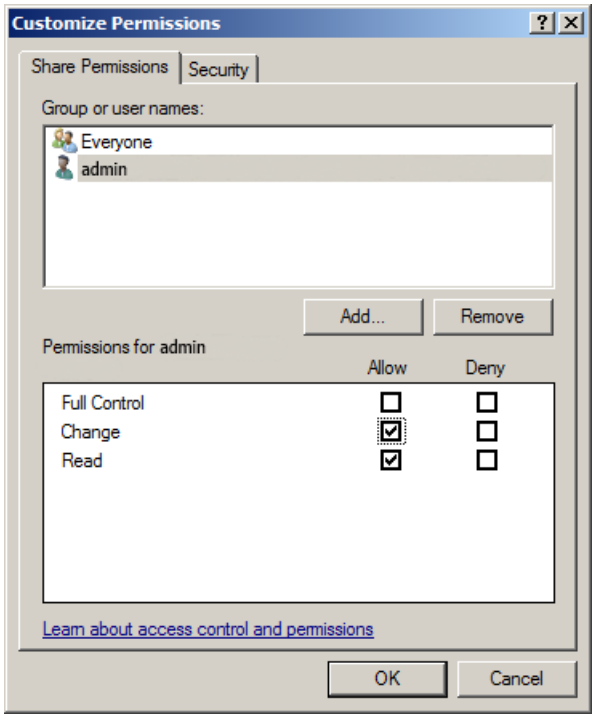



Figure 6-16 "Customize Authorizations" dialog box

6. In section "Group or user names," click "Add..."
   The "Select Users, Computers, Service Accounts, or Groups" opens.

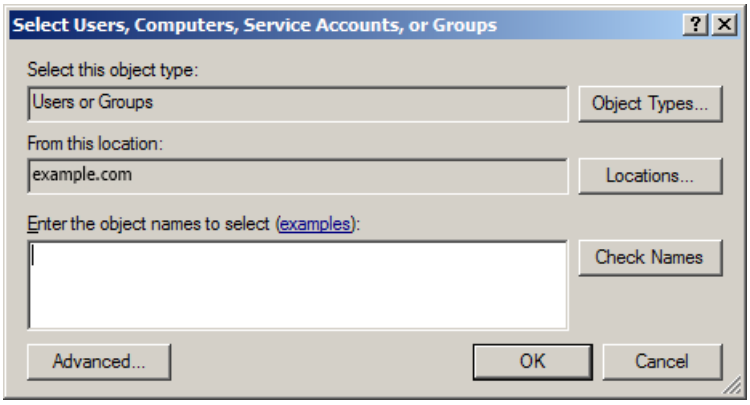



Figure 6-17 "Select Users, Computers, Service Accounts, or Groups" dialog box

7. In section "Authorizations for <user name>," under column "Allow," click the "Change" checkbox.
8. Confirm these settings in the "Customize PermiAuthorizations" dialog box with "OK."
9. In the next steps of the wizard, make all the settings you require and confirm the share with "Finish".

## 6.12.6 Creating a shortcut to the network drive

To be able to access the shared directory on the PG/PC from the Siemens PC system, after you have connected and configured the PG/PC in the network, you must set up the folder released on the PC system as a network drive.

As soon as it has been set up, a new drive will appear on the PC system, which is really a directory on the PG/PC.

The basic procedure for setup under Windows 7 and Windows XP is described below. Detailed information from Microsoft is provided in Windows under "Help and Support" or on the Internet: Creating a shortcut to a network drive (association) (http://windows.microsoft.com/en-us/windows7/create-a-shortcut-to-map-a-network-drive)

### Requirement

- The PG/PC is switched on.
- The Service Desktop is active.
- The folder on the PG/PC is shared for the relevant user accounts on the domain and on the PC system.

## Procedure

Proceed as follows to connect a shared directory as a network drive:

1. Start the Windows Explorer and click the "Map network drive" button.
2. Make the following settings in the "Map network drive" dialog box:

| Settings | Purpose |
|---|---|
| Drive | Select drive letters under which the network directories are to be accessible on the PC system.<br><br>The drive letters Z, Y and X are assigned to USB interfaces on SINUMERIK TCUs and operator panels. Do not use these drive letters for other purposes in order to avoid problems when using these USB interfaces. |
| Folder | Specify the device in the network and share name of the resource.<br>Syntax: `\\Name of the server\share name of the network directory`<br>Example: `\\Backup_Server\PCU_Backup` |
| Reconnect at logon | Activate this checkbox to automatically connect the network drive on booting. |
| Connect using different credentials | Activate this checkbox if you have not shared the directory in the network with the user for whom you would like to set up the network drive.<br>In the next dialog box "Enter Network Password," enter the log-on information of a user with whom the network directory is shared. |

3. Confirm your settings with "Finish".

## See also

Microsoft Windows support (http://windows.microsoft.com/en-US/windows/windows-help#windows=windows-7)

## 6.13 Starting up the replacement SSD

Information on the replacement or installation of an SSD can be found in the hardware documentation for your PC system:

- SINUMERIK PCU 50.5
  - Manual SINUMERIK 840D sl Operator Components and Networking
- SIMATIC IPC
  - SIMATIC IPC627 operating instructions
  - SIMATIC IPC427 operating instructions

**Note**

**Troubleshooting if reading in of the disk image fails**

If data transfer is interrupted during the reading process, there will be no executable operating system available on the Siemens PC system.

In this case, use a bootable USB flash drive with Service Center to read the disk image in again.

### See also

Network settings (Page 10)

Restore a disk image of the SSD (Page 130)

# Service and diagnostics 7

## 7.1 PCU Hardware Diagnostics

### Intended use

The PCU hardware supports the diagnostics of important system components via an integrated "safecard", which is designated as a Safecard-On-Motherboard (SOM).

### Monitored Parameters

The following physical parameters of the PCU hardware are monitored:

- CPU temperature
- Housing temperature
- I/O chip temperature
- Speed of the two housing fans
- S.M.A.R.T - status of the hard disk

### Logging faults

The PCU hardware monitor logs all hardware faults in the Windows event log so that the faults can be output even without installing an HMI software.

The alarms are output in the log under "Control Panel" → "Administrative Tools" → "Event Viewer".

### See also

- Starting up the replacement SSD (Page 153)
- Operator Components and Networking Manual: Chapter, spare parts
- Description of the alarms: SINUMERIK 840D sl Diagnostics Manual

# 7.2 Evaluation of the diagnostic displays H1/H2

The PCU 50.5 has the diagnostics displays "H1" and "H2", which output status codes by means of LEDs and 7-segment displays. In particular, these simplify the diagnostics of the PCU 50.5 in headless mode (without a screen or input devices).

If an error is found, you can connect a screen for further error analysis.

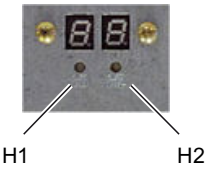



Figure 7-1 Diagnostics displays H1 and H2 of the PCU 50.5 with LEDs and 7-segment displays

The diagnostics displays display the following information:

- The diagnostics display "H1" displays status codes of the PCU-Basesoftware.
- The diagnostics display "H2" is assigned to the application software.

The meaning of the displayed status codes depends on whether the PCU 50.5 is in system startup (Page 156) mode, running mode (Page 156), or shutdown (Page 156) mode.

## Further information

Additional information on the diagnostics display, e.g. the meaning of the BIOS POST codes, can be found in the Manual SINUMERIK 840D sl Operator Components and Networking.

Table 7-1 Status codes of the diagnostics display during the system power up

| LED H1 | LED H2 | 7-segment display H1/H2 | | Status | |
|---|---|---|---|---|---|
| Orange | Orange | Output of the BIOS post codes | | After switching on the system | |
| Orange | Off | | | After cycling the BIOS | |
| Orange | Off | | | After starting Windows | |
| | | 1 | 0 | Starting Windows | Loading of the device drivers required by Windows |
| | | 2 | 0 | PCU hardware service | The PCU hardware service has been started. |
| | | 5 | 0 | Network | Wait for network interfaces to be ready |
| | | 8 | 0 | TCU support test step 1 | Wait for FTP server to start |
| | | 9 | 0 | TCU support test step 2 | Wait for TCU network boot process and starting of the TCU hardware service |
| | | A | 0 | TCU support test step 3 | Wait for VNC server to start |
| | | B | 0 | - | - |
| | | E | 0 | PCU-Basesoftware | General fault: see Windows "Event Viewer" |

Table 7-2 Status codes of the diagnostics display during operation

| LED H1 | LED H2 | 7-segment display H1/H2 | | Status | |
|---|---|---|---|---|---|
| Flashing green | -- | 0 | 0 | OK | |
| Flashing red | -- | | | Error: | |
| | | 1 | 0 | Temperature alarm | Housing or CPU temperature; temperature above limit value (SOM error) |
| | | 2 | 0 | Fan alarm | Housing or CPU: Low fan speed or fan failure (SOM error) |
| | | 3 | 0 | Hard disk alarm | S.M.A.R.T hard disk error |
| | | 6 | 0 | | VNC server failure or VNC server service stopped |

Table 7-3 Status codes of the diagnostics display during the shutdown process

| LED H1 | LED H2 | 7-segment display H1/H2 | | Status |
|---|---|---|---|---|
| Flashing red/ green | -- | 5 | 0 | Shutdown in progress |

# 7.3 Setting of the operating mode during run-up

## 7.3.1 Service switch (PCU 50.5)

The service switch on the PCU 50.5 is a rotary switch with 16 possible positions.

The "PCU Hardware Service" component as part of the PCU-Basesoftware provides a function to read-out the switch positions.

The switch position on the hardware can be used to specify the operating mode for the run-up of the PCU 50.5.

You can find the meanings of the switch positions under Possible operating modes (Page 159).




Figure 7-2 Service switch

### Further information

You can find details on the hardware of the PCU 50.5 in the *Manual SINUMERIK 840D sl Operator Components and Networking.*

### 7.3.2 Possible operating modes

Only the settings of the service switch that are relevant for the commissioning engineer are listed:

| Switch setting | Function |
|---|---|
| **0** | Normal mode (switch setting in delivery condition) |
| **3** | Desktop operation:<br>Run-up in the "SINUMERIK" splash screen which enables the desktop selection.<br>Relevant if HMI software is installed with autologon mode set.<br>See also: Switching to the Service Desktop during autostart / autologon operation (Page 160). |
| **9** | Reset BIOS to factory settings |
| **E** | Boot device at the USB interface, e.g. USB flash drive with emergency boot system<br>See also: Bootable USB flash drive (Page 140) |

### 7.3.3 Software-side setting (SIMATIC IPC)

In contrast to SINUMERIK PCUs, SIMATIC IPCs have no service switch on the hardware side for selecting the mode. Therefore, on the SIMATIC devices, you can define the mode on the software side in the configuration file`pcuhwsvc.ini`.

#### Archiving of the configuration file pcuhwsvc.ini

The template is located in directory `C:\ProgramData\Siemens\MotionControl\siemens\System\etc\`

Do not overwrite this template, but save a copy of the template `pcuhwsvc.ini` in one of the user directories:

- `C:\ProgramData\Siemens\MotionControl\`**user**`\System\etc\`
- `C:\ProgramData\Siemens\MotionControl\`**oem**`\System\etc\`
- `C:\ProgramData\Siemens\MotionControl\`**addon**`\System\etc\`

You can find detailed information on the configuration files in Chapter: Directory structure and file conventions (Page 14).

Table 7-4 Mode setting in configuration file pcuhwsvc.ini

| | |
|---|---|
| Section | [GLOBAL] |
| Key | `ModeSwitch=` |
| Value | 0 = Normal mode<br>3 = Desktop mode (startup in the welcome screen)<br>Other modes (Page 159) are not permitted in the software-side setting for SIMATIC IPC. |
| Factory setting | 0 |

## 7.4 Switching to the Service Desktop during autostart / autologon operation

If you have set up the autologon operation and the autostart of SINUMERIK Operate with the activated keyboard filter, the PC system runs up with SINUMERIK Operate after being switched on.

To be able to switch to the Windows desktop in this case, interrupt the startup of SINUMERIK Operate with an entry, as described here.

Alternatively, you can switch the mode for the next startup on the hardware side (SINUMERIK PCU 50.5) (Page 158) or on the software side (SIMATIC IPC) (Page 159).

### Requirement

- SINUMERIK Operate is installed and autologon mode is set up.
  See Chapter Setting up SINUMERIK Operate for autologon mode (Page 98).
- The PCU is switched off.

## Procedure

To switch to the desktop when the autostart/autologon mode is activated, proceed as follows:

1. Switch on the PC system and wait until the "SINUMERIK" welcome screen is displayed.
2. While the "SINUMERIK" welcome screen is being displayed, make one of the following entries:
   - Press the key 3 on your keyboard.
   - Click or tip the copyright information.

   The counter for starting SINUMERIK Operate is suspended and you can select one of three options:




Figure 7-3 Welcome screen "SINUMERIK" after key input

3. Select the "Display Desktop" option.

## Result

The Service Desktop opens.

# 7.5 Configuring the SINUMERIK power up screen

You can parameterize various settings to change to the service desktop or to the SINUMERIK welcome screen. These settings can be especially helpful if you wish to carry out diagnostics and maintenance to a system in the headless mode.

## Precondition

- In the corresponding user directory, file `slstartup.ini` is either not available, or there is no `slstartup.ini` with different settings in a user directory with a higher priority.
  If the file already exists in the corresponding user directory, then do not create a new `slstartup.ini`, but instead, adapt the existing file or insert section `[StartScreen]` using the appropriate keys.

## Procedure

Proceed as follows to create the configuration file for the direct control key layout:

1. Open a text editor, and as character coding, set ASCII or ANSI.
2. Insert the section designation:
   `[StartScreen]`

3. Enter those keys, whose values you wish to parameterize, and make the required settings:

| Setting/key | Possible values | Description |
|---|---|---|
| ShowButtons= | [true\|false] | Specifies as to whether the buttons are displayed with power up options directly in the "SINUMERIK" welcome screen.<br>For "ShowButtons=false", the buttons are only displayed after making a selection (key 3 or click on the Copyright note). |
| Timeout= | s | Time in seconds, after which the system powers up with SINUMERIK Operate.<br>The counter starts to run after an operator panel, screen or a VNC connection has been identified. |
| MaxTimeout= | s | Maximum time in seconds, after which the system powers up with SINUMERIK Operate.<br>The counter starts to run immediately, irrespective of whether an operator panel, screen or a VNC connection exists. |
| ShutdownTimeout= | s | Time in seconds, within which the system power down can be interrupted (key 3 or click on the Copyright note). |
| ShowCountdown= | [true\|false] | Displays a countdown until the system powers up with SINUMERIK Operate. |

4. Save the file as `slstartup.ini` in one of the user directories.
   - `C:\ProgramData\Siemens\MotionControl\user\sinumerik\hmi\cfg\slstartup.ini`
   - `C:\ProgramData\Siemens\MotionControl\oem\sinumerik\hmi\cfg\slstartup.ini`
   - `C:\ProgramData\Siemens\MotionControl\addon\sinumerik\hmi\cfg\slstartup.ini`

   See also: Directory structure and file conventions (Page 14)

## Example

**Default setting of the section [StartScreen] in startup.ini**

```
[StartScreen]
ShowButtons=false
Specifies as to whether the buttons are displayed with power up options directly in the "SINUMERIK" welcome screen (without key 3 or clicking on the copyright note).
Timeout=5
# Time in seconds, after which the system powers up with SINUMERIK Operate (assuming that a screen was identified).
MaxTimeout=300
# Maximum time in seconds, after which the system powers up with SINUMERIK Operate (without a screen having been identified).
ShutdownTimeout=3
```

**Default setting of the section [StartScreen] in startup.ini**

```
# Time in seconds, within which the system power down can be interrupted
(key 3 or click on the Copyright note).
ShowCountdown=true
# Displays a countdown until the system powers up with SINUMERIK Operate.
```

# 7.6 Remote access

## 7.6.1 Overview

You can use a remote connection to access other devices via the network:

- Access to maintenance with prompt (SSH) via PuTTy.
- Access via VNC Viewer with all of the operating and servicing options, which are also provided directly on the Siemens PC system.

### See also

Setting up and using SSH (Page 169)

## 7.6.2 Searching for devices in the system network

The "sc_show_net" command shows devices available in the system network in the prompt and offers additional information about these devices.

As both these functions rely on SNMP (Simple Network Management Protocol), only SNMP-capable devices will be found. In the case of Windows-based devices, this depends on the software version. If a simple call is made without any additional options, a list of the devices found will appear showing the relevant IP address, DNS name (if known), and a short description (module name).

### sc_show_net command

This command is executed in the prompt:

| | |
|---|---|
| Syntax: | sc_show_net [-xml] [[-hw] [-tco] [-sw \| -swfull] [-loc] [-panel] [-dhcp] [-switch] \| -all] [HOSTS...] |
| Authorization level: | None |

## 7.6.3 Display of accessible stations in SINUMERIK Operate

You can use the network diagnostics function in SINUMERIK Operate to show accessible nodes in the system network.

### Procedure

Proceed as follows to show accessible nodes:

1. Select "MENU SELECT" to open the menu.
2. In the horizontal softkey menu, choose "Diagnostics > Bus TCP/IP."
3. In the vertical softkey menu, choose "TCP/IP > Diagnostics Network > Accessible nodes." The "Accessible nodes" screen opens showing all nodes that use the SNMP or DCP protocol.

### Further information

You will find Information about showing accessible nodes in SINUMERIK Operate in the *SINUMERIK 840D sl SINUMERIK Operate (IM9)* Commissioning Manual, Section *Service and diagnostics*.

## 7.6.4 Remote access to the PCU for operation and maintenance

You can connect via remote access to devices which have an active VNC server, e.g. PCU 50.5 or NCU.

- The remote access to the Siemens PC system from a PG/PC in the system network (Page 141) is activated via the factory setting. You can optionally define a password for this:
  If you want to access the PC system via the company network instead or in addition, you must adapt the firewall settings (Page 72) and define a password.

  **Note**

  **Access to the PC system in the company network requires a secure password**

  A password is required to access the PC system via remote access in the company network. The password is not yet defined in the factory settings. If the PCU is to be accessed in the company network, you must specify a password:

- With the factory settings, a system (e.g. PCU 50.5) can only be monitored via an external VNC Viewer.
  To operate the system from another station, the system in question must grant permission. You make the necessary settings in the **tcu.ini** in section `[VNCViewer]`.

### Defining the password

To define passwords for remote access on the PC system, call the prompt and enter the "sc_vncpwd" command.

| Command | Parameter | Value |
|---|---|---|
| sc_vncpwd | set | companynetwork\|systemnetwork [password] |
| | reset | companynetwork\|systemnetwork |
| Authorization level: None | | |

- Example for company network (X1):
  sc_vncpwd set companynetwork mypasswd
- Command for system network (X2):
  sc_vncpwd set systemnetwork mypasswd

The password must not be more than 8 characters long. Passwords are saved in encrypted form in the following file: C:\ProgramData\Siemens\MotionControl\user\System\etc \sinumerikvnc.ini

**Note**

**Specification of the password is also required for remote access to an NCU (without PCU/IPC) in the company network**

As from SINUMERIK Operate V4.8 SP2 a password is required for access via VNC to an NCU in the company network. The password is not yet defined in the factory settings. If the NCU is to be accessed in the company network, you must specify a password as follows:

1. Call the command line of the NCU (TCU menu "Select service session > Show command shell of NCU") or putty.
2. Log in as user "manufact".
3. Execute the "sc vncpwd set companynetwork mypasswd" command, whereby "mypasswd" is a password with a maximum of eight characters.

Further information is displayed for the input of "sc vncpwd".

## Further information

- Further information on the permissible network structures and network configurations can be found in the Operator Components and Networking Manual.

## Options for the VNC viewer

**Note**

**Do not change the factory settings!**

To ensure that the VNC viewer works properly, the following options must not be changed.

Table 7-5 Connection settings in the start dialog of the VNC Viewer

| Option | Factory setting |
|---|---|
| VNC Server | 192.168.214.241 |
| Quick Options | AUTO (Auto select best settings) |

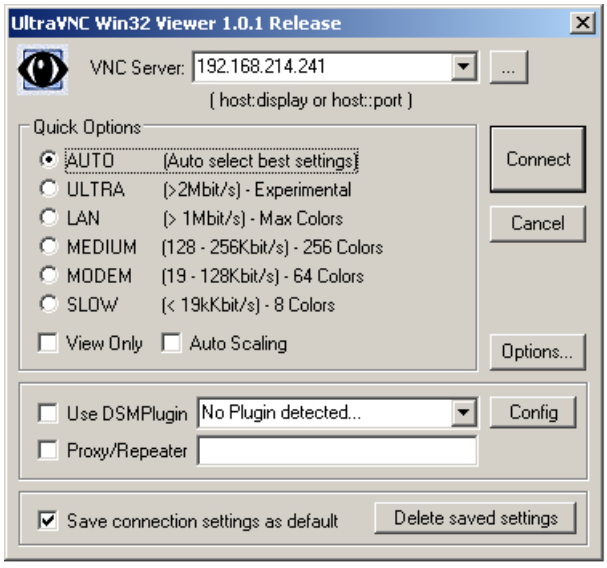
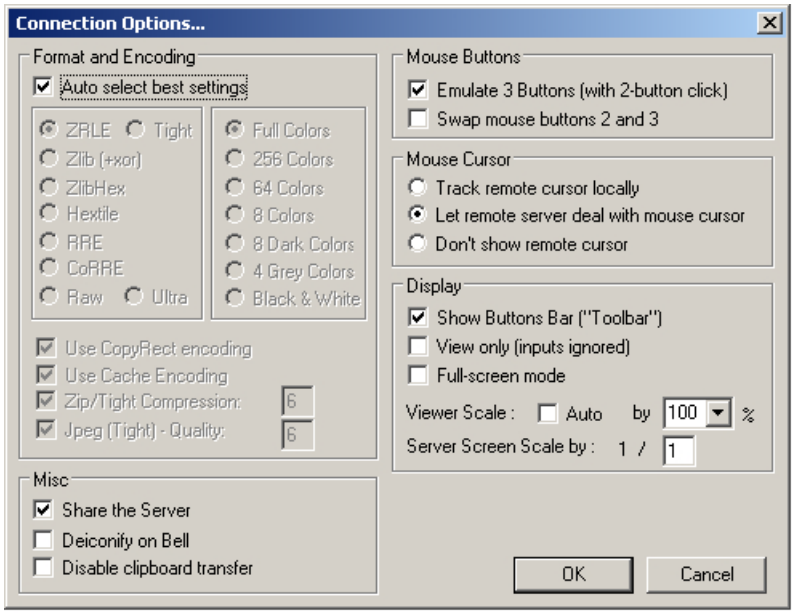



Table 7-6 Further connection settings ("Options..." button)

| Option | Factory setting |
|---|---|
| Format and Encoding | Auto select best settings |
| Misc | Share the Server |
| Mouse Buttons | Emulate 3 Buttons (with 2-button click) |
| Mouse Cursor | Let remote server deal with mouse cursor |
| Display | Show Buttons Bar ("Toolbar") |



## 7.6.5 Setting up and using SSH

You can use remote access to access devices from the PCU.

An SSH environment is preinstalled when the PCU base software for Windows 7 is installed. You can log on with PuTTy.

You can use an SSH key to log on. A script for setting up the infrastructure on devices is provided. However, for security reasons no previously generated keys have been stored.

### Setting up remote access with SSH and key pair

The procedure for setting up users and using remote access is the same as the usual procedure for remote access with SSH.

Table 7-7 Support with setting up remote access

| Task | Recommended application | Storage path | Help |
|---|---|---|---|
| Generate key pair | PuTTy Key Generator | C:\Program Files (x86)\PuTTy \puttygen.exe | Online help in PuTTy (Chapter *Public key for pasting into authorized_keys file*) |
| Set up infrastructure on devices | Supplied script ssh_key_login.bat | C:\ProgramData\Siemens\Motion Control\siemens\etc. \ssh_key_login.bat | Call up the script via the prompt. To do this, enter user name and file name of the public key as the parameters.<br>**Syntax:**<br>`ssh_key_login.bat <user name> <public key>`<br>**Example:**<br>`ssh_key_login.bat username publicKey.pub` |
| Access devices via remote connection | PuTTy | C:\Program Files (x86)\PuTTy \putty.exe | Online help in PuTTy<br>IP address of the PCU (factory setting):<br>• System network: 192.168.214.241<br>• Company network: IP address is taken from the DHCP server |

### See also

Encryption via SSH protocol (Page 170)

Remote access (Page 165)

## 7.6.6 Encryption via SSH protocol

### Security

The security of SSH is ensured by a series of cryptographic algorithms for encryption and authentication.

### Authentication

The server identifies itself to the client with an RSA, DSA or ECDSA certificate, which enables detection of manipulation in the network (no other server can identify itself as a known server).

The client can authenticate itself either by public key authentication with a private key whose public key is stored on the server, or with a normal password. While in the latter case, user interaction is always necessary (unless the password is stored unencrypted on the client computer), public key authentication allows client computers to log on to SSH servers without user interaction, without a password having to be stored in plain text on the client. However, for added security the private SSH keys can also be protected with a password.

### Subsystems

In the case of Secure Subsystem Execution, subsystems that were defined in an SSH server installation can be executed remotely without knowing the precise path of the program executed on the server. SFTP is the most common subsystem.

### Further information

- Chapter Setting up and using SSH (Page 169).
- "NCU Operating System" Commissioning Manual (IM7).
- Description of PuTTy.

# Index


## 7

7-segment display, 156

## A

Answer file, 20
Autologon, 20
    Switch to Service Desktop, 160
Autologon mode, 100
AutoRepeat, 102

## B

Backup time, 59
Batch file
    USVShutdown.bat, 52, 53, 56
Batch processing file, (Batch file)
BIOS
    BIOS post code, 156
    Factory setting, 159
Boot stick, 140

## C

Color depth, 42
Command, (Command)
    control userpasswords2, 101
    sc_show_net, 165
    sc_usb disable, 33
    sc_usb enable, 33
Command line, (Prompt)
Commissioning
    unattended, 20
    USB flash drive, 20
    with identical settings, 20
Company network
    Defining the password, 166
Configuration file
    Archive, 14
    basesys.ini, 34, 144
    ghost.ini, 62
    oemframe.ini, 107
    pcuhwsvc.ini, 102, 159
    PCUInst.ini, 108, 110, 111, 112
    servicesystem.ini, 62, 63, 64
    setup.ini, 26
    sinumerikvnc.ini, 166
    slstartup.ini, 162
    tcu.ini, 37, 38, 42, 43, 166
    Templates, 14
    vdck.ini, 47, 48

## D

DCK, 45
DHCP, 10, 12
Diagnostics display, 156
Dialog box
    netplwiz, 100
    ServiceCenter Backup/Restore, 62, 63, 64
Direct control key, 45
Directory
    C:\ProgramData, 14
    D:\Install, 20
Disk image
    Restoring, 130, 134
    Troubleshooting, 131, 136
DOS shell, (Prompt)

## E

Emergency Boot System, 140, 159

## F

File and printer release, 34
FirstLogonCommands, 20

## H

Hard disk
    Replacing, 153
HMI software, 107

## I

Initialization files, (Configuration file)
Input area scheme, (Keyboard layout)
Installation, (See Commissioning)
Interface
    Ethernet, 10, 12

X1, 10, 12
X2, 10, 12

## K

Key, (Repeat function)
Key filter, 102
Key filter, 102
Keyboard layout, 35, 98
Kiosk mode, (Autologon mode; key filter)

## L

Language
Configuring the keyboard layout, (Keyboard layout)
Language bar, (Keyboard layout)
Log File, (Log file)
CBS.log,
PCUHardwareservice.log,
PCUInstaller_C.log,
PCUInstaller_S.log,
Setup-specific,

## M

Mode of operation
Headless mode, 20
Multitouch, 43

## N

Network
Adapting settings, 32, 63, 64
Company network, 10, 12
Domain, 32
IP address, 32
System network, 10, 12
Network protocol
SNMP, (SNMP)
Network release
Driven on PG/PC, 148
Network security
Define password for VNC server, 166

## O

OEM
Configuration, 107
Operating mode, 159
Autologon mode, (Autologon mode)
Service switch, 158

## P

Password, (Password)
Administrator account, 20, 30
Authentication for SSH encryption, 170
Configure for VNC server, 166
For network drive, 137
For unattended installation, 20
Password reset diskette, 30
unattend.xml, 20
PCU
Checking the hard disk, 155
PCU Installer
Activating, 108, 110
Configuring, 111, 112
Log file, (Log file)
PCUInst.ini, 112
Troubleshooting, 110, 117
Using, 116
Port 80 display, (See 7-segment display)
Power-up
Operating mode during run-up, 159
Prompt
Open, 33

## R

Remote access, 166
Via VNC server, 166
With SSH encryption, 169
Response file
unattend.xml, 23

## S

Screen resolution, 37, 38
Service desktop, 162
Service switch, 158, 159
Service system for PCU, 140
Shutdown, 51
SINUMERIK Operate
Unattended installation, 20
SITOP UPS, 51
Softkey, 107
SSD
Replacing, 153
SSH encryption, 169, 170

Startup
Initial startup, 20
STEP 7, 107

## T

Troubleshooting
PCU Installer, 110, 117
Restore failure, 131, 136

## U

Uninterruptible power supply, 51
USB flash drive
Deactivate network access, 34
with emergency boot system, 140
USB interface
Activating, 33
Disabling, 33
User account, (Setting up for autologon mode)
Setting up, 29, 30
Setting up for autologon mode, 98
User administration, 30

## V

VDCK, 45
Virtual direct control keys, 45
VNC Viewer, 166

## X

XML
Response file, 20
unattend.xml, 20
XML file
unattend.xml, 20, 23

# Appendix

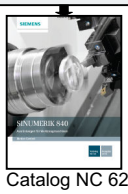
# A

## A.1 Documentation overview SINUMERIK 840D sl

**General documentation**



Advertising brochure


Catalog NC 62


Catalog PM 21



System Manual
Ctrl-Energy



System Manual
Smart machine tool operation

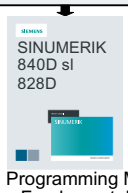
**User documentation**



Operating Manual
– Universal
– Turning
– Milling
– Grinding



Create MyConfig
Operating Manual



Programming Manual
– Fundamentals
– Job Planning
– Measuring Cycles



Programming Manual
– ISO Turning
– ISO Milling



Diagnostics Manual,
Alarms

**Manufacturer/Service documentation**



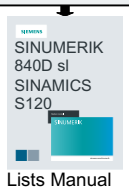
Equipment Manual
– NCU
– Operator components and networking
– ADI4



System Manual
Guidelines for machine configuration



Commissioning Manual
– CNC: NCK, PLC, Drive
– Base software and operating software



Lists Manual
– Machine data
– Interface signals
– Variables

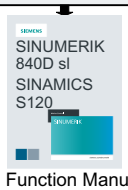
**Manufacturer/Service documentation**



Function Manual
– Basic functions
– Extended Functions
– Special Functions
– Synchronized Actions
– ISO Dialects



Function Manual
Tool management

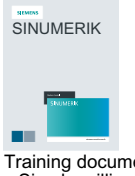


Function Manual
Safety Integrated



Configuration Manual
– EMC installation guideline
– Industrial security

**Info / Training**



Training documentation
– Simple milling with ShopMill
– Simple turning with ShopTurn



Tool and Mold Making Manual

**Electronic documentation**


DOConCD


Industry Online Support (SIOS)


Industry Mall