SIEMENS

SINUMERIK

SINUMERIK 840Di sl/840D sl/840D Base software and HMI Advanced

Commissioning Manual

Valid for:

Controls
SINUMERIK 840Di sl/840DiE sl
SINUMERIK 840D sl/840DE sl
SINUMERIK 840D/840DE

Software version:
HMI Advanced V7.6

03/2009
6FC5397-0DP10-3BA0

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

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

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 adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the 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
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Ordernumber: 6FC5397-0DP10-3BA0
℗ 01/2009

Preface

SINUMERIK documentation

The SINUMERIK documentation is organized in three parts:

- General documentation
- User documentation
- Manufacturer/service documentation

An overview of publications, which is updated monthly and also provides information about the language versions available, can be found on the Internet at:

http://www.siemens.com/motioncontrol

Follow menu items "Support" → "Technical Documentation" → "Ordering Documentation" → "Printed Documentation".

The Internet version of DOConCD (DOConWEB) is available under:

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

Information about training courses and FAQs (Frequently Asked Questions) can be found at the following website:

http://www.siemens.com/motioncontrol under menu item "Support".

Target group

This documentation is intended for commissioning personnel.

The plant/product is installed, connected, and ready to start. The Commissioning Manual ought to contain all necessary information about or at least references to subsequent procedures such as testing the cabling, power on, and functional testing.

Benefits

The intended target group can use the Commissioning Manual to test and commission the product/system correctly and in total safety.

Utilization phase: Setup and commissioning phase

Standard scope

The functionality of the standard scope is described in the following documentation. The machine manufacturer documents supplements or changes that he makes (the machine manufacturer).

Other functions not described in this documentation might be executable in the control. However, no claim can be made regarding the availability of these functions when the equipment is first supplied or in the event of 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.

Questions about the documentation

If you have any queries (suggestions, corrections) in relation to this documentation, please fax or e-mail us:

Fax	+49 9131 98 2176
E-mail	docu.motioncontrol@siemens.com

A fax form is available at the end of this document.

Technical Support

If you have any technical questions, please contact our hotline:

	Europe/Africa
Phone	+49 180 5050 222
Fax	+49 180 5050 223
Internet	http://www.siemens.com/automation/support-request

	America
Phone	+1 423 262 2522
Fax	+1 423 262 2200
E-mail	techsupport.sea@siemens.com

	Asia/Pacific
Phone	+86 1064 719 990
Fax	+86 1064 747 474
E-mail	techsupport.asia@siemens.com

Note

For technical support telephone numbers for different countries, go to:

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

Calls are subject to charge, e.g. €0.14/min. from a German landline. Tariffs of other telephone service providers may vary.

SINUMERIK Internet address

http://www.siemens.com/sinumerik

EC Declaration of Conformity

The EC Declaration of Conformity for the EMC Directive can be found on the Internet at:

http://support.automation.siemens.com

under the Product Order No. 15257461 or at the relevant branch office of the I DT MC Division of Siemens AG.

CompactFlash cards for users:

- The SINUMERIK CNC supports the file systems FAT16 and FAT32 for CompactFlash cards. You may need to format the memory card if you want to use a memory card from another device or if you want to ensure the compatibility of the memory card with the SINUMERIK. However, formatting the memory card will permanently delete all data on it.
- 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.
- If you cannot use a memory card with the SINUMERIK, it is probably because the memory card is not formatted for the control system (e.g. Ext3 Linux file system), the memory card file system is faulty or it is the wrong type of memory card.
- Insert the memory card carefully with the correct orientation into the memory card slot (take note of arrows, etc.). 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 the SINUMERIK keeps to the 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).
- The CompactFlash card from SanDisk "CompactFlash® 5000 Industrial Grade" has been approved for SINUMERIK (Order Number 6FC5313-5AG00-0AA0).

SIEMENS

SINUMERIK

SINUMERIK 840Di sl/840D sl/840D HMI Advanced (IM4)

Commissioning Manual

Valid for:

Controls
SINUMERIK 840Di sl/840DiE sl
SINUMERIK 840D sl/840DE sl
SINUMERIK 840D/840DE

Software version HMI-Advanced 7.6

03/2009
6FC5397-0DP10-3BA0

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
with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION
without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE
indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

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 adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the 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
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Ordernumber: 6FC5397-0DP10-3BA0
℗ 01/2009

Table of contents

Introduction 1

1.1 State of the system when supplied

Overview

This manual describes the commissioning of the HMI-Advanced software. When commissioning the SINUMERIK control, you may require additional manuals:

- Operator Components and Networking
- Commissioning Manual PCU Basic Software
- Diagnostics Manual
- Parameter Manual
- Function Manual Basic Functions

  Additional information on special NCK, HMI, PLC or drive functions are provided in the Function Manuals.

Software

Depending on the order details, the HMI-Advanced software may be loaded on the PCU 50.3 at the time of delivery. If the HMI-Advanced software is not installed on the PCU 50.3 (e.g. PCU 50.3 is supplied without system software), the HMI software can be installed on the PCU 50.3 from the CD through Service Center.

To install the HMI-Advanced software on the PCU 50.3, you need the following:

- PC or PG with CD drive
- Network connection
- USB memory

The HMI Advanced software can run on the Windows XP SP2 operating system.

NOTICE
Installing on a PC/PG: HMI-Advanced can then only be run under a non-administrator user, if this user has write authorization for directory mmc2 of HMI-Advanced. The installation of HMI-Advanced does not provide this authorization.

1.2 Booting

1.2.1 Settings at the HMI

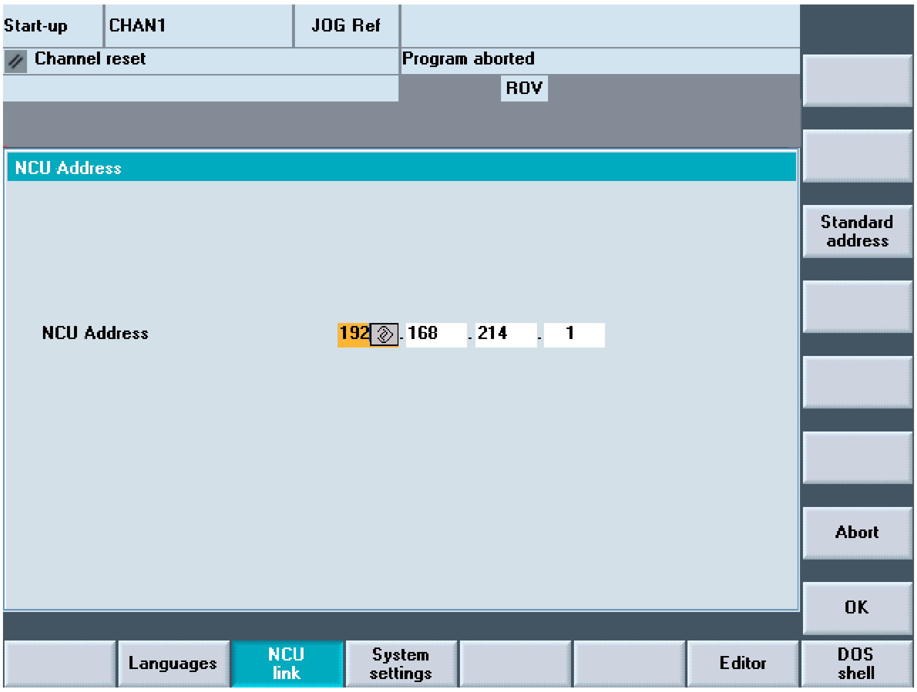
Selecting languages

Once you have selected a first and second language in the "Startup" area, you can subsequently toggle between them using the "Change Language" softkey. The following standard languages are available:

- German
- English
- French
- Italian
- Spanish
- Simplified Chinese

NCU link (solution line only)

This function allows you to adjust the IP address of the NCU. The entered IP address saved in the file MMC.INI is displayed.



Start-up | CHAN1 | JOG Ref

Channel reset | Program aborted

ROV

NCU Address

NCU Address 192 . 168 . 214 . 1

Figure 1-1 IP address of the NCU (for solution line)

The NCU is supplied from the factory with the default address 192.168.214.1. In the case of a 1:1 link, this address can be maintained without the need for additional networking.

Pressing the “Default address” softkey transfers the factory set IP address 192.168.214.1 to the address field for the NCU.

However, if the control is linked to a company network, for example, the IP addresses will be different. You must restart the HMI for the changes to take effect. The section with the new IP address is written to user/mmc.ini.

Error when booting

The following error can occur while booting:

```
Alarm 120202: Waiting for a connection to the NC/PLC
```

Explanation:	This alarm occurs if the HMI program is started for the first time and the NCK/PLC has not yet finished booting or if communication with these components is faulty. When this alarm occurs, all display values connected with NCK/PLC become invalid. Such faults are normal while the controls are starting up (e. g. after resetting).
Remedy:	The alarm disappears automatically as soon as the fault situation is resolved. If this alarm persists, the cause of the fault may be one of many (for example, wire break, NCK/PLC has not booted, incorrect address/baudrate configuration, ...).
Responses:	--

System settings

See the next chapter, System settings.

Printer selection

The softkey only works if at least one printer is installed under Windows. It can be used to print displays or data from the commissioning operating area. You can use the <SELECT> button to choose which of the installed printers should be used for output.

Default: Output as bitmap file

Editor

This key opens the ASCII editor in which files can be edited at Windows level.

You can select existing drives via the vertical softkeys.

See also

Configuring user alarms (Page 86)

1.2.2 System settings

Overview

Under "System settings", settings are made for inquiry windows, file tree display and screen display in the Machine, Program and Services operating areas.

File display

You can set the file tree display for the Services, Machine and Program operating areas. The following columns can be selected:

- File type (extension)
- Downloaded
- Length
- Access protection
- Date
- Time
- Enable
- Display levels (branch to directory trees, max. 7)
- Name length (max. 25 characters)

Sorting information

A dialog box is displayed for setting the sorting sequence of **one** column. The defined sequence applies when displaying the corresponding window in the Machine, Program and Services operating areas of the selected column.

You can select from the following sort criteria:

- Without sorting: Sorted in ascending order according to "Name" by default.
- One of the column designations: The sorting sequence runs in ascending or descending order. Confirm with OK.

The sorting sequence set is shown as an arrow symbol next to the name of the column selected as the sorting criterion. In HMI-Advanced with an optional mouse, the following operating options are also available for column sorting:

A click on the column name with the arrow symbol changes the direction and sorts the information accordingly. A click on another column makes this the sorting criterion. Click again to change the direction, if necessary, as in (1.). The selection of another sorting criterion in the operating area changes the sorting criterion for all operating areas (Machine, Program, Services).

Note

When the sorting sequence is set, if the sorting criterion is not available in the display image of the operating area, the information is sorted in ascending order according to the column name. The sorting sequence defined in the dialog applies to the operating areas, in which the criterion set in the dialog appears.

System queries

You can specify whether or not an inquiry window should be displayed after certain commands, e.g. Delete. Request acknowledgement:

- Deleting data/programs,
- Deleting directories,
- Overwriting files.

The "Representation of keys in displays..." window is opened: Here you can define whether keys must be represented as icons or as text in HMI displays. Example: Operator panel front in US layout

Use workpiece templates

When creating a new workpiece, you can specify here whether templates should be transferred to the new workpiece (directory):

- Job lists
- Part programs
- Initialization programs

Trace

In the event of communication errors, a trace log for communication processes can be recorded following an instruction to the Service department or to our hotline. The trace log is only evaluated by Siemens.

HMI exit mode

During the commissioning phase it is often necessary to shut down the HMI-Advanced and operating system and then reboot the PCU. To avoid powering-down and powering-up the PCU or machine, select “Automatically reboot control on shutdown”.

See also

Configuring the prompt dialog box for the EXIT mode (Page 23)

1.2.3 Behavior of the keys for the PCU

CAPSLOCK

The CAPSLOCK function allows all entries made with **external** SINUMERIK keyboards to be in upper case instead of lower case. If lower case letters are needed, they can be added by means of the SHIFT commands function.

This function is only active when HMI-Advanced is running. It is not active in operation with Windows NT/XP.

Restrictions

When the CAPSLOCK function is activated, the <SHIFT> key has no effect on letter keys.

As is the case in standard Windows applications, CTRL/ALT key sequences only work in lower case mode. Switching to lower case mode is not available on keyboards that are integrated in the OP.

The <CTRL> + <SHIFT> switching sequence only works with external PS2 keyboards and USB PC keyboards, not with panel keyboards.

Note

The OP keyboard enters lower case text if CAPSLOCK is set on an external keyboard. If the external keyboard is unplugged in this state, the CAPSLOCK mode cannot be cancelled.

The <CAPSLOCK> and <NUMLOCK> keys can be filtered by a corresponding setting in parameter file E:\Windows\System.ini.

The default setting for CAPSLOCK and NUMLOCK is not filtered.

Activation

This function is activated by display MD 9009: $MM_KEYBOARD_STATE

0:	CAPSLOCK off
2:	CAPSLOCK on

This display MD is only evaluated when booting. If the machine data is changed, the new setting only becomes effective after booting.

The display MD 9009 is set to "CAPSLOCK on" and you want to enter lower case letters:

1. Press <CTRL> + <SHIFT> keys to switch to lower case letters.
2. To reset text entry from lower case back to upper case, press <CTRL> + <SHIFT> again.

1.3 Access levels concept

Overview

The access levels concept controls access to functions and data areas. Access levels 0 to 7 are available, where 0 represents the highest level and 7 the lowest level.

Access levels 0 to 3 are locked using a password and 4 to 7 using the appropriate key-operated switch settings.

Access level	Locked by	Area
0	---	System
1	Password: SUNRISE	Manufacturer
2	Password: EVENING	Servicing
3	Password: CUSTOMER	Users
4	Key-operated switch setting 3	Programmer, machine setter
5	Key-operated switch setting 2	Qualified operator
6	Key-operated switch setting 1	Trained operator
7	Key-operated switch setting 0	Semi-skilled operator

The password remains valid until it is reset with the "Delete Password" softkey. The password for access level 0 provides access to all data areas.

The passwords can be changed after activation. If the passwords have been forgotten, for example, the system must be reinitialized (NCK general reset). This resets all passwords to the standard for this software version. POWER ON does not reset the password.

Key-operated switch

Access levels 4 to 7 require a corresponding key-operated switch setting on the machine control panel. Three keys of different colors are provided for this purpose. Each of these keys provides access only to certain areas. The associated interface signals are located in DB10.DBB56.

Significance of the key-operated switch settings:

Access level	Switch setting	Key color
7	0 = Key removal position	No key inserted
6-7	0 and 1	Black
5-7	0 to 2	Green
4-7	0 to 3	Red

Authorization

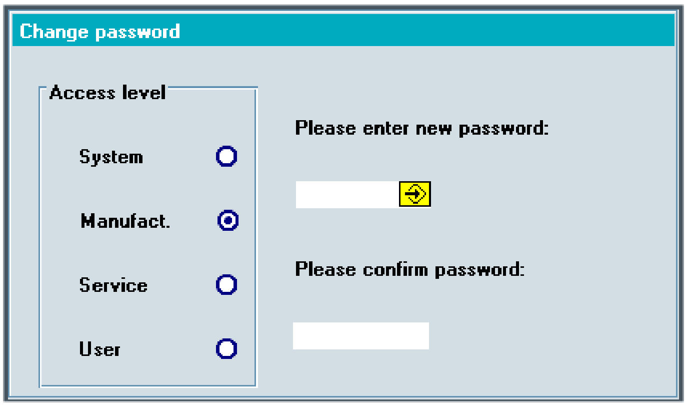
The operator only has access to information protected by this particular access level and the levels below it. The machine data is assigned different access levels by default.

Access level 4 (key-operated switch position 3) is the minimum level required to display machine data. When commissioning the system, the manufacturer password "SUNRISE" should be generally used.

Changing the password

Procedure:

1. Press the "Password" softkey.
2. Press the "Set password" softkey.
3. The input window is displayed "Please enter password"
4. Enter one of the three possible passwords and press the Enter key or icon to acknowledge. A permissible password is acknowledged as set and the currently applicable access level is displayed. Invalid passwords will be rejected. You must delete the old password before activating a password for a lower access level than the one activated.
5. Press the "Delete password" softkey.
6. After pressing the "Delete password" softkey the valid password is deleted and deletion is acknowledged. The currently valid access level: Key-operated switch setting 0 is set.
7. If a password is already set, then you can change it as follows. Press the "Change password" softkey. The "Change password" input box appears:



Change password

Access level

System ○

Manufact. ●

Service ○

User ○

Please enter new password:

Please confirm password:

Figure 1-2 Changing the password

8. The modified password must be entered in both input fields then confirmed by pressing the "OK" softkey. Both entered passwords must match for the modified password to become valid.

1.4 Licensing (solution line only)

Licensing an option

The HMI-Advanced options manager provides support when entering the License Key for an option.

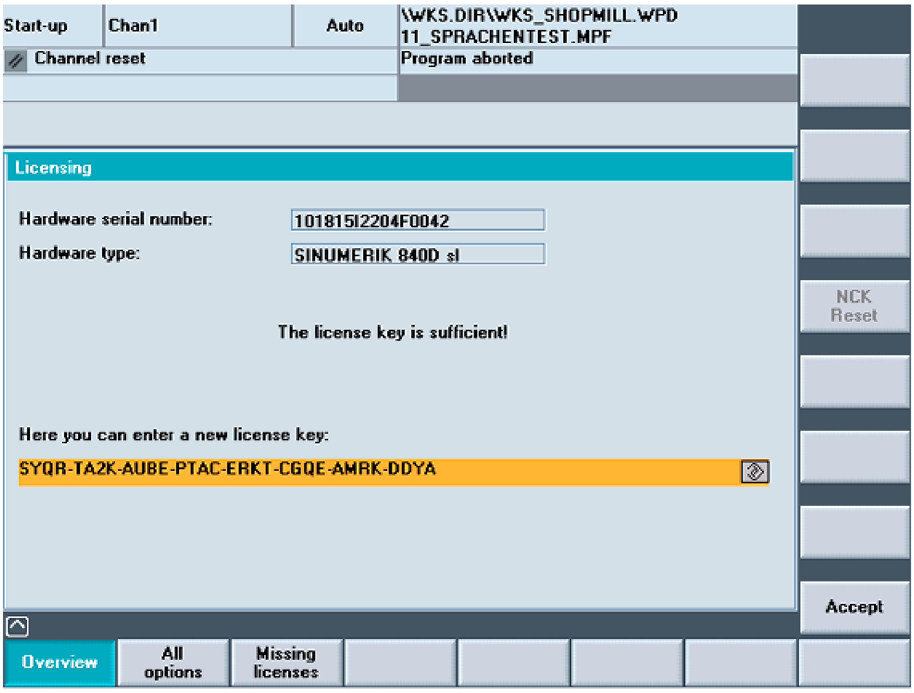
Note

Default setting

The option data for the NC is subject to a protection level for retailers/end users.

The input dialog for setting the options and inputting the license key is assigned protection level 3 = end user.

After input of a license key, the "Accept" softkey is activated. After you have pressed the "Accept" softkey, the license key is written into the NCK where it is checked for validity.



Start-up | Chan1 | Auto | \WKS.DIR\WKS_SHOPMILL.WPD
11_SPRACHENTEST.MPF

Channel reset | Program aborted

Licensing

Hardware serial number: 101815I2204F0042

Hardware type: SINUMERIK 840D sl

The license key is sufficient!

Here you can enter a new license key:

SYQR-TA2K-AUBE-PTAC-ERKT-CGQE-AMRK-DDYA

Figure 1-3 Licensing overview

If you have entered an invalid license key it will be rejected by the NCK and a message will appear. If an incorrect license key is entered on three occasions, an NCK power-on/reset is required.

Note

In NCK the need for a power ON/Reset depends on the procedure when commissioning:

- If the option bit is first set and then the function is commissioned, then a reset is required after commissioning.
- However, if the function is commissioned first, a reset initiated and then the options bit is set in the options manager, a reset must be carried out again to activate the changes.

No reset is required after writing the license key, the "License key set!" message appears in the user response line.

References: Manual SINUMERIK 840Di sl, Chapter "Automation License Manager"

All options

All of the options selectable for this control are listed here. Furthermore, the list states whether or not a valid and adequate license key has already been entered and how many licenses are available.

Missing licenses

This displays the options that are already set, which are in the current license key but not yet licensed. In HMI the options are only checked when booting. Consequently, HMI must be rebooted in each case.

1.5 Limits of data management

Overview

Data are stored in the following directories:

- Workpieces
- Part programs
- Subroutines
- User cycles
- Standard cycles
- Manufacturer cycles

The data management directories can contain a combined total of 100,000 files, whereby each directory (each *.WPD workpiece directory for workpieces) is limited to 1000 files. Files in other directories do not count toward the total limit of 100,000 files. However, each of these directories is also limited to 1000 files, e.g. a maximum of 1000 archives in the archive directory. The following limit applies to network drives: maximum of 1000 files per directory.

In practice, the number of files available also depends on the file sizes and the available memory space. A large number of files slows down the display generation for directory displays.

A total of 5000 customized alarms or messages can be created in each language.

2 Configuring the HMI system

2.1 Processing INI files

Notation

[xxx]	;Designating a section
Identifier=value	;Value assignment
Identifier=<empty>	;Delete value
,	;Separator (comma) for multiple assignments
;	;Introduction of comment: As a general rule, comments indicate the range of values
REM	;Introduction of comment

Directory structure

The system directory structure is organized in such a way that user changes are retained if the software is upgraded.

- User changes to the original software are made in parallel directories.
- There is a clear separation between standard HMI software and customer-specific enhancements.
- The "mmc2" and "hmi_adv" directories (and their subdirectories) should be read-only directories. They contain the original software supplied. These directories are overwritten only if the software is upgraded.

The diagram below shows the priority sequence. This means that: Entries in the directory to the right overwrite corresponding entries in directories to the left.

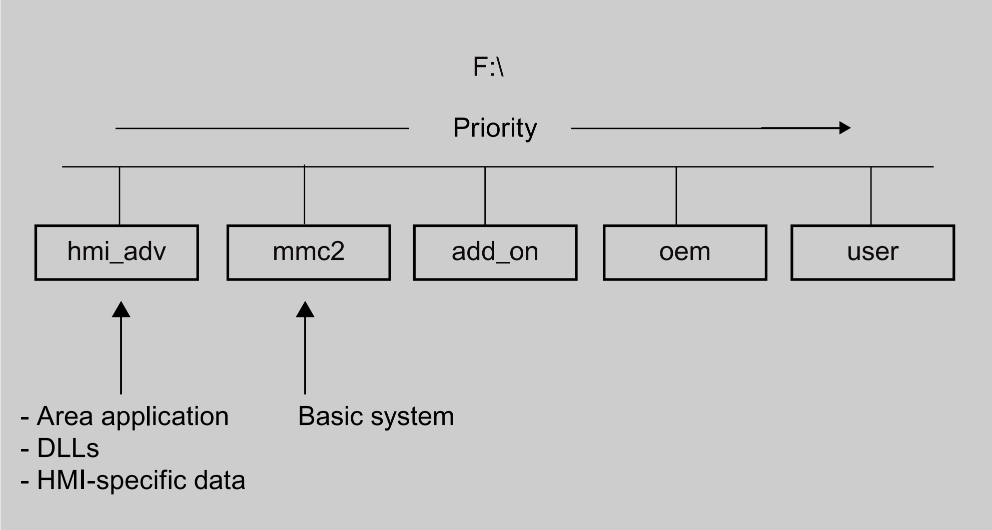


Figure 2-1 Priority

Contents of the directories

As a general rule, only **entries that differ** from the originals in mmc2 should be stored in the parallel directories for mmc2.

NOTICE
None of the INI files in the "mmc2" and "hmi_adv" directory may be modified.

The directories contain the following contents:

- hmi_adv:

  The directory contains area applications, DLLs and HMI-specific data.
- mmc2:

  System directory of the HMI software.
- add_on:

  Directory for additional Siemens products (e.g. TPM, MDA, DNC ...). This can have the same directory structure (e.g. subdirectory \language) and subdirectories as mmc2. Entries for an add-on product in REGIE.ini and re_*.ini, for example, are also stored here.
- oem:

  Directory for machine manufacturers and OEM users in which internal OEM applications are located. This can have the same directory structure (e.g. subdirectory \language) with subdirectories as mmc2. Entries for an OEM product in REGIE>INI and re_*.ini, for example, are also stored here.
- user:

  User directory in which customer differences from the ini files supplied are stored. **Changes** to the appearance of the user interface, which can be made by the user by means of settings on the HMI user interface, are also stored here (e.g. language setting, file selection, file manager view, etc.). Other settings that do not concern OEM or add-on products are also stored in the directory \user, e.g. alarm server settings.

Changes to “auxiliary files”, e.g. assignment of specific access authorizations for functions, or hiding of specific softkeys, also have to be stored in the directory \user.

Example:

A complete file REGIE.INI is not generated in the directory "oem", but only the modified sections are saved.

REGIE.INI would then look like this, for example:

```
[TaskConfiguration]
Task7:=oem1appl, .....
```

Diagnostics when an error occurs

The settings of the INI files are analyzed and displayed using the HMI analyzer.

Procedure for user-specific entries

In the F:\USER directory, create an empty INI file with the same name as in "mmc2" (if necessary, also create subdirectories, e.g. \language, in USER).

In the INI file in the USER directory, copy only the section you wish to change and write the new or modified entry below it.

Note

Do not copy the entire INI file from "MMC2"! Only copy the differences.

Example:

Entry in the file \USER\MBDDE.INI for PLC error messages and scroll in the alarm line:

```
[TextFiles]
UserPLC = F:\DH\MB.DIR\MYPLC_

[Alarms]
 ;Scroll in the alarm line
RotationCycle = 1000
```

The same procedure applies to the F:\ADD_ON and F:\OEM directories. \ADD_ON is reserved for Siemens products. All OEM applications should be installed in the directory \OEM.

NOTICE
The associated entries, e.g. in REGIE.INI, should be modified in these directories and **not** in \MMC2\!

Switching off settings

If, for the search sequence, there are specific entries in a lower-priority directory that you wish to be disabled in a directory with a higher-priority, you can do this by specifying an <empty> entry instead of a value.

Example:

Directory mmc2, file xxx.INI:

```
[<Section>]
<Identifier> = <value>
```

Directory user, file xxx.INI:

```
[<Section>]
<Identifier> = <empty>
```

Maximum file length

The maximum file length for INI files in Windows systems is 60 kbytes. If this limit is exceeded, entries at the end are ignored.

Remedy: Omit the full comments in \USER\PARAMT.INI, for example, since they are also included in MMC2\PARAMT.INI.

2.1.1 Terminate OEM commissioning

Function

This function transfers the initialization data from the USER directory to the OEM directory. This provides the user with an empty USER directory and deletes any existing initialization files (*.ini) stored there, without losing the settings it contains because the initialization files are taken out of the USER directory and put in the OEM directory. If no OEM directory is available one is created automatically, if initialization files are stored in the USER directory.

Combine all INI files:

1. If you wish to accept **all** initialization files, press the "Terminate OEM st.-up" softkey. The following message will appear: "Combine all initializing files (*.ini) from the USER directory with the relevant files of the OEM directory."
2. Press the "Save" softkey to start data transmission.

   Existing entries in the OEM directory are overwritten by the corresponding entries from the USER directory.
3. New files are created.

   Files which do not appear in the USER directory, but already exist in the OEM directory, are saved. During transmission the name of each file is displayed in the status bar. Once all of the data has been transmitted successfully the file is deleted from the USER directory.

Accept selected INI files:

If you wish to only accept specific initialization files, press the “Select data” vertical softkey. A list of all files stored in the USER directory is displayed.

2.1.2 Activating the screensaver

Function

Tasks of the screensaver:

- HMI screensaver: Protection for screen and backlighting
- Windows screensaver: Protects the screen

The delay time in [min] until the screensaver is activated, is set in the MMC.INI file in the section [GLOBAL]:

```
[GLOBAL]
; latency for the screensaver
MMCScreenOffTimeInMinutes = 60
```

NOTICE
The HMI screensaver and Windows screensaver must not be used at the same time.

Displaying server names

In order to make server names visible, activate the following entry:

```
[GLOBAL]
; to make the hidden servers ( NCDDE, MBDDE, DHSERVER, ARSERVER ) visible, uncomment this entry
ServerVisible = 1
```

2.1.3 Configuring the prompt dialog box for the EXIT mode

Function (only for Windows XP, not on PG/PC)

This function enables HMI-Advanced (including Windows XP) to be shut down followed by a reboot of the PCU during the commissioning phase. This enables you to avoid having to power-down/power-up the PCU or the machine.

The entry can be set either directly in the REGIE.INI file or the user interface via Startup → HMI → Settings → HMI Exit Mode. If the “EXIT” menu is configured accordingly, the **"Shutdown... "** or **"Restart ... "** prompt appears depending on the access stage.

Settings in the REGIE.INI file:

Default: No prompt dialog box

```
[Miscellaneous]
EnableRebootDialog = True
```

If the protection level is lower than the manufacturer level (protection level 0 to 2) or if the above entry is set to "False" in the REGIE.INI file, the prompt can be confirmed with:

OK	HMI is closed and Windows XP is shutdown.
Cancel	HMI is not closed

If you have the applicable access authorizations (protection level 3 to 7), the following option is offered:

Restart	HMI and Windows XP are closed and then restarted.
Exit	HMI is closed and Windows XP is shutdown.
Cancel	HMI is not closed

2.1.4 Setting the storage location for the alarm log

Function

Two different methods are available saving the alarm profile on the hard disk. The application and the disk load must be taken into account when selecting which method to use. The required entries are made in the MBDDE.INI file.

Writing to alarm files always places a load on the same hard disk region. Various methods can be chosen with the following control options.

There are a number of strategies for reducing the load on the disk. Multiple log files can be maintained in parallel on the disk. With the multiple file strategy, the next file is selected each time the HMI is booted. This reduces the load on the disk hardware, both in the data area as well as the directory information area. In addition, this strategy detects physical disk errors in the data area on startup and avoids them by reallocating memory space.

The multiple file strategy can also be selected using the following entry in the file mbdde.ini.

```
[PROTOCOL]
DiskCare
```

The following values are possible:

DiskCare=-1	The MBDDE server controls the alarm log in the memory. The alarm log is saved on the hard disk if it is displaced in the operator area "Diagnostics" of if the key <Alarm Cancel> is pressed.
DiskCare=0	The data is written to the log file immediately.

DiskCare=n	Changes of the alarm state are written to the log file if no change occurs in n seconds. In addition, the following applies: The alarm log is saved on the hard disk if it is displayed in the operator area "Diagnostics" of if the key <Alarm CANCEL> is pressed.
DiskCare=-n	n>1 specifies the number of parallel alarm files.

The file "mbdde.ini" is evaluated when booting. The unused alarm files and the current log file are located in the mmc2 directory. The names of these hidden files (attribute "hidden") consist of an 8-digit hexadecimal number with the extension ".alr". The names of files in which write errors are detected when booting also consist of an 8-digit hexadecimal number, but with the extension ".al_".

Log file size

The size of the log file (ring buffer) can be defined in the MBDDE.INI file.

```
[Alarms]
Records = value ; size of log file
```

Default: 150	
Minimum value: 18	With default value of 150, 75 alarms can be displayed. 2 records are needed for each alarm.
Maximum value: 32000	

Selecting alarms by feature

Filter entries in MBDDE.INI can be used to control the messages logged in the log file.

```
[PROTOCOL]
Filter=Expression
```

This expression is based on the following syntax:

Syntax

```
[IDENTIFIER][RELATION][FEATURE][OPERATORS]
```

- IDENTIFIER:

No.	Alarm number
Prio	Priority
Mode	Message line/alarm line or dialog
Type	Alarm type (Power On, CANCEL, etc.)
From	Source of alarm
AckVar	Acknowledgment variable

- RELATION:

":"	Equal to
"<"	less than
">"	greater than
"!"	Not

- FEATURE: Numbers of strings
- OPERATORS

","	Comma denotes logical OR, **only within** a filter
" "	Space/blank denotes logical AND **between** individual filters.
"|"	Pipe denotes logical OR **between** individual filters.

Examples:

Filter=Type<3	Only Power On-and RESET alarms are logged.
Filter=From:NCU_1	Only alarms from NCU_1 are logged.
Filter=From:NCU_1 Type:1,3	Only Power On and CANCEL alarms from NCU_1 are logged.

See also

MBDDE.INI (Page 298)

2.1.5 Configuring the acknowledgement icon for PLC alarms

Function

You can configure your own acknowledgement icon for acknowledging PLC alarms. You can choose which bitmap is displayed in the diagnostics alarm image for PLC-generated alarms (via FC 10). The chosen pixel image (20 x 20 pixels) should tell the user which button to press to acknowledge alarms generated by DB2.

The chosen button must correspond to the signal specified in the QUIT acknowledgement parameter by FC 10. For example,

```
Call FC 10
ToUserIF:=TRUE
Quit:=DB21.DBX7.7 //Channel 1 RESET
```

Select a value for the image to be displayed in the [ALARM_PICTURE] section:

```
[ALARM_PICTURE]
;0 = Image for PLC button
;1 = Image for Cancel button
;2 = Image for Reset button
;3 = Image for HMI button
;4 = Image for User Defined button
ButtonImage=1
```

If 4 is selected (User Defined button), then the corresponding image must be defined as ButtonNameUser="*name*.bmp". Any value can be chosen for *name*.

We recommend making the entry in the DG.INI in the USER or OEM directory.

The image must be located in one or more of the following directories:

;F:\User\Icons\640

;F:\User\Icons\800

;F:\User\Icons\1024

;F:\Oem\Icons\640.

;F:\Oem\Icons\800.

;F:\Oem\Icons\1024.

It is automatically adjusted to the current screen resolution. A version of the image must be located in the directory containing the images in the current resolution.

2.1.6 Changing access levels for programs

Function

When booting, the data management server checks whether the DH.INI file contains entries for modified standard access levels. The data management server uses these entries to update its internal image of the data management chart with the modified access levels.

The access levels described in the entries are now used as standard access levels for the selected data. Entries that do not match the existing scheme are ignored.

Restrictions

When the HMI database is installed, data is always created with the access levels defined in the standard data chart. Access levels for nodes such as wks.dir cannot be changed.

Formats and data type

All entries are located in the [ACCESSMASKS] section. The entries have, e.g. the following format:

```
[ACCESSMASKS]
```

\wks.dir*.wpd*.mpf =	75775
\mpf.dir*.mpf =	75775
\cus.dir*.spf =	33773

A standard access mask can be declared for each data type and storage location. An entry comprises the path at which data of the data type can be created and the new standard access mask.

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

The new standard access mask 75775, with which new part programs are created, is defined in all workpieces.

Access mask 75775 stands for:

7	Read access for all	0 to 7 are permissible
5	Write authorization for protection level 5 and higher	0 or 7 are permissible
7	Execution authorization is set	0 or 7 are permissible
7	Display authorization for all	0 or 7 are permissible
5	Delete authorization for protection level 5 and higher	0 to 7 are permissible

See also

DH.INI (Page 280)

2.1.7 Configuring the "'Language Selection" softkey

Function

In order to select more than 2 languages on the control, the "Language Selection" softkey can be configured in the main screen "Start-up" in order to change the language using the menu.

The entry "SkSelectLngInsteadSkChangeLng" in the IB.INI file in the "[LANGUAGE]" section controls whether the "Change Language" softkey in the main Start-up screen should be replaced by the "Language Selection" softkey. If there is no "SkSelectLngInsteadSkChangeLng" entry, then only "Change language" is available.

The values of "SkSelectLngInsteadSkChangeLng" have the following significance:

False:	"Change Language" softkey for changing between the foreground and background language, same as before (default setting).
True:	"Select Language" softkey to select one of the installed languages. This means that online, more than two languages can be changed over at the language change position after the appropriate selection.

If the setting is TRUE (Language Selection), the following values should be added to the entries in the "[LANGUAGE]" section in the MMC.INI file:

```
LanguageList= ... , GR
FontList= ... , Europe
FontListKO= ... , Europe2
LBList= ... , German
```

Note

With Language Selection the first language is not provided as a choice. If you list the first language again in the language list (and therefore also in the corresponding font settings), then the first language is also listed in the language selection menu.

Assigning parameters

The names of the softkeys are predefined.

```
HSx x 1 - 8 , horizontal softkeys 1 to 8
VSy y 1 - 8 , vertical softkeys 1 to 8
Not all softkeys need to be assigned.
```

References: Commissioning Manual "Supplementing the operator interface", Chapter "Configuring OP hotkeys and PLC keys".

2.1.8 Creating user operator menus

Function

Skeleton applications can be generated with the following **additional options**:

- Free design of operating tree: Creation of new operating levels, new arrangement of softkeys for previous operating levels and the new operating levels.
- Integration of user applications in the operating levels (OEM applications).
- Calling standard applications from any operating level, including defined submenus.

Using the "Skeleton application" function, existing operating areas used in HMI-Advanced can be combined and selected with a new softkey – simply by means of **configuration.** This releases operating areas/softkeys in the main menu for dedicated operating areas (OEM applications).

The "Skeleton application" function can be used repeatedly to define a subordinate operating level from an existing operating level. This allows free operating tree design.

A skeleton application can define up to 16 horizontal and 8 vertical softkeys. When you leave an operating area that was selected from a skeleton application, you return to the higher-level menu of the skeleton application.

The following standard applications can be selected from a skeleton application:

- Machine
- Parameters
- Services
- Program
- Simulation → HMI settings
- Diagnostics
- Commissioning
- OEM application(s)

From the skeleton application, a particular function of an operating area that is accessible from it can be selected specifically. (e.g. Program → Simulation, all standard possibilities are specified in the table below). A skeleton application can provide its own main screen as a bitmap file or display a standard image.

Configuration principle

New operating levels are created by skeleton applications and attached to existing operating levels. The assignment of horizontal and vertical softkey bars to skeleton applications can be configured.

Configuration files

Configuration is carried out in the following files:

REGIE.INI	Calls the skeleton application(s), standard applications In REGIE.INI an interpreter task mntmmc is specified as a task in which the CmdLine:="SectionName1" parameter indicates the section in which additional information about the inserted skeleton application in the TASKCONF.INI file is located.
TASKCONF.INI	Description of softkeys and the operating areas and background screen/texts they call, texts for the softkey labeling, if required. The specifications are made under the section that was specified in the CmdLine attribute.
RE_xx.INI	Softkey labeling of the skeleton application for the language with the code XX.

Example

The following entries in the **REGIE.INI** clarify the process:

Skeleton application for **user-specific operating trees**.

```
; sample entry
Task7 = name := mntmmc, cmdline := "SectionName1", Timeout := 60000
The task number (7 in this case) is assigned to the softkey that
; calls the general application.
 Task0: Horizontal softkey 1
...
Task7: Horizontal softkey 8
```

mntmmcis the name of a standard task for interpreting the softkeys in the skeleton application in **TASKCONF.INI** and for activating them.

The value of **cmdline** indicates the section in the TASKCONF.INI file in which the softkey assignments in the skeleton application are described in detail.

There may be more than one Task<No.>= name := **mntmmc**, ... entry if several sections in TASKCONF.INI have been defined with skeleton application parameters. This also allows operating trees to be designed with multiple levels.

```
[SectionName1]
; REGIE.INI must contain a reference to the section name selected by
the user.
;
; Sample entries:
; The Services application should be activated when the general
application
; is started for the first time.
; The softkey index is defined using the StartIndex entry
; that activates the required task when the program starts.
; Services with StartIndex 5 for example
StartIndex = 5
; Main screen for the skeleton application
Picture= Skeleton1.bmp
; The first softkey is assigned the task program (Taskindex 2)
; as child (HSK1IsTask=0) with parameter <root>
; the softkey texts for German are specified via default and English
(UK)
;
HSK1Task=2
HSK1IsTask=0
; application-specific parameters: Here, entry into the
; basic level of the application program, parameters: <root>
HSK1Command=<root>
HSK1SkText=Program
; No language specified is the default setting for all
; languages that are not configured
HSK1SKText _UK=Program
; softkey label in English (_UK)
; The fifth softkey is assigned the task services (Taskindex 3) as
; child (HSK1IsTask=0) with the empty command parameter, that
; specify the softkey texts for default German (_GR) and English
(_UK).
; is predefined.
HSK5Task=3
HSK5IsTask =0
HSK5Command=
HSK5SkText=SERVICES
HSK5SkText _GR=Services
HSK5SkText _UK=service
; Access authorizations for the "Services" softkey
HSK5AccessLevel= 3
; the 2nd softkey of the ETC bar is assigned the commissioning task
; (Task index 5) as child(HSK10IsTask=0)
; with an empty command parameter,
; that specifies the softkey texts for German (_GR) and English
(_UK).
;
HSK10Task=5
HSK10IsTask=0
HSK10Command=
HSK10SkText _GR=commissioning
HSK10SkText _UK=setup
; TerminateTask=
; HSK16=ExitButton
```

Assignment of softkey numbers

The following assignments apply to HSK and StartIndex:

1 - 8: Horizontal softkeys 1 - 8

9 - 16 horizontal softkeys in the ETC softkey bar

17 - 24 vertical softkeys 1 - 8

25 - 32 softkey bar for OEM applications

Language-dependence of softkeys

The softkeys are labeled according to the current language setting. The softkeys are updated whenever the language is changed. If the softkey text for the current language (**HSK**<SoftkeyIndex>**SkText_**<Language>**=**...) is not found, the softkey is assigned the default text (Entry **HSK**<SoftkeyIndex>**SkText=**...) or, if there is no default text, the task index.

Skeleton application as child task

The skeleton application can also be started from a dedicated process as a child application. Then, upon recall, the skeleton application returns to the starting application and closes, if applicable:

; The application is terminated on return

HSK<SoftkeyIndex>TerminateTask=1 ; <> 0 end

or

; The application is not terminated on return. This is the default setting !

HSK<SoftkeyIndex>TerminateTask=0 ; 0 do not end

Main screen for the skeleton application

Unless otherwise specified with Picture= …, a blank background is displayed.

You can specify your own picture and store it as a bitmap in the **oem** directory, e.g. Picture=BackgroundSkeleton1.bmp

The screen is displayed when the skeleton application is selected or on return from an application called by the skeleton application.

Entering 1 rather than a picture file causes the "Start-up" main screen to be displayed (default).

Note

If StartIndex and a picture are configured, the picture is superimposed as soon as the skeleton application is selected and the application configured with StartIndex opens. If the picture is larger than the window available in the current operating device, the part from the center of the picture that fits in the window is displayed.

Access authorizations

The softkeys in the skeleton application can be secured with access authorizations.

```
0: System
... ...
7: Key-operated switch setting 0
; Example:
; Access authorizations for the "Services" softkey
```

HSK5AccessLevel= 3

NCK-dependent softkeys

With HSK<SoftkeyIndex>NckLink=1, a softkey can be tied to the existence of an intact connection to the NCK, provided softkey entries in TASKCONF.INI are made accordingly. The softkey can then only be used when the connection exists. Default is 0: no check.

Shortened configuration in TaskConf.INI

The above configuration of TASKCONF.INI for a softkey in the skeleton application can be simplified for the following entries when standard operations are addressed:

```
HSK<SoftkeyIndex>Task= x/y
x operator area
y command or state= i
```

Example:

```
HSK1Task=Program/<root>
```

has the same priority as the full configuration:

```
HSK1Task=2
HSK1IsTask=0
HSK1Command=<root>
HSK1SkText=Program
```

Table 2- 1 Assignment of the tasks and softkey texts

x	y	Explanation
Machine	<root>	Entry to the Machine main screen **(task 0)** with recall to the highest level for return. The softkey text is *Machine*.
Parameters	<root>	Entry into the Parameter main screen **(task 1)** with recall to the highest level for return. The softkey text is *Parameter*.
Program	<root>	Entry into the Program main screen **(task 2)** with recall to the highest level for return. The softkey text is *Program*.
Service	<root>	Entry into the Service main screen **(task 3)** with recall to the highest level for return. The softkey text is *Services*.
Diagnostics	<root>	Entry into the Diagnostics main screen **(task 4)** with recall to the highest level for return. The softkey text is *Diagnostics*.
Diagnostics	State=10	Entry into the Diagnostics main screen with recall to the highest level for return and selection of the alarm screen. The softkey text is *Alarms*. **Note:**The entire horizontal diagnostics softkey bar can be used.
Diagnostics	State=20	Entry into the Diagnostics main screen with recall to the highest level for return and selection of the message screen. The softkey text is *Messages*.
Diagnostics	State=30	Entry into the Diagnostics main screen with recall to the highest level for return and selection of the alarm log screen. The softkey text is *Alarm log*.
Diagnostics	State=40	Entry into the Diagnostics service screen with recall to the highest level for return. The softkey text is *Service displays*.
Diagnostics	State=50	Entry into the Diagnostics PLC status screen with recall to the highest level for return. The softkey text is *PLC status*.
Setup	<root>	Entry into the **Start-up (task 5)** main screen with recall to the highest level for return. The softkey text is *Start-up*.
Setup	State=10	Entry into the Machine data screen with recall on exiting the level. The softkey text is *Machine data*.
Setup	State=40	Entry into the PLC overview screen in Start-up with recall on exiting the level. The softkey text is *PLC*.

x	y	Explanation
Setup	State=50	Entry into the Drive overview screen in Start-up with recall on exiting the level. The softkey text is *Drives/Servo*.
IBSetup	<root>	Entry into the **Settings (task 34)** main screen with recall to the highest level for return. The softkey text is *System settings*.
Simulation	<root>	Entry into the **Simulation (task 27)**. A modal dialog is displayed, which you can use to select the program to be simulated. The simulation is exited by pressing Cancel. The softkey text is *Simulation*.

Other interfaces in the standard applications

The following standard applications can be addressed with these:

- Parameters
- Diagnostics
- Commissioning

The following commands should be noted to the right of the equals sign in the softkey command definition:

HSK<SoftkeyIndex>**Command**=Command1; Command2; ...

or

VSK<SoftkeyIndex>**Command**=Command1; Command2; ...

A semi-colon is used as a separator between multiple commands.

Example:

; Hide some softkeys in the horizontal softkey bar for standard operation

HSK1Command=DisableHSK(1, 3-4)

Commands

DisableHSK(<*softkey list*>)

The horizontal softkeys are removed from the *softkey list*. Individual softkeys can be separated by commas or specified using softkey ranges <*from*> - <*to*>, e.g. DisableHSK(1,3,5-7).

Note

The softkeys in the ETC bar cannot be removed.

DisableVSK(<*softkey list*>)

The vertical softkeys are removed from the *softkey list*. Individual softkeys can be separated by commas or specified using softkey ranges <*from*> - <*to*>, e.g. DisableHSK(1,3,5-7).

DoVSK=<*SoftkeyIndex*>

The action of the vertical softkey (1-8) is initiated.

Recall(<*status list*>)

On reaching a status from the list the application returns to the calling application. The statuses can be separated by commas or specified as ranges <*from*> - <*to*>, e.g. Recall(1,5-7,48).

Highlight=<*Softkeyindex*>

The horizontal (1 – 8) or vertical (9 – 16) softkey is displayed with a blue background when the application is called.

ZuMat=<*status matrix*>

The status matrix (e.g. dg\dg.zus) transferred is read and replaces the original status matrix.

RE_GR.INI

```
[HSoftkey texts]
HSK7 = "Special menu" // 20
```

The creation of OEM applications in HMI-Advanced is described in:

Reference: HMI programming package

See also

Transferring the actual task number of the HMI to the PLC (Page 116)

TASKCONF.INI (Page 327)

2.1.9 Supplementing service displays on a user-specific basis

Function

The DGOVW.INI file is used to define additional signals in drive diagnostics. The file must be created explicitly in the mmc2 directory or addon, oem, user directory.

If the file is present, the signals in it are evaluated and displayed in the operator area "Diagnostics" → "Service displays" → "Service overview". Language-specific and non-language-specific texts can be configured in it.

Language-specific texts are located in language-specific INI files in the mmc2\language (or addon, oem, user...) directory with the name DgOvwTxt_XX.ini.

A log file called dgovw.log is created in the user directory, in which error messages that occur when the dgovw.ini file is compiled are output.

Structure of the DGOVW.INI file

```
[GLOBAL]
NrOfSignals=

[BMP]
100="<Bitmapname>"
101="..."
...
 [SIG1]
text=
Item=
expr=
.
.
[SIG2].
...
 [SIG<n>]
...
```

Significance of entries

`NrOfSignals:`

Number of additional signals. The signals are located in the [SIG<nr>] section, where <nr> goes from 1 to NrOfSignals (continuously and with no gaps).

`Text:`

language-specific text in the form $T<TextNr>

where <TextNr> is a number in the language-specific DgOvw_xx.ini file (xx is the language abbreviation).

non-language-specific text "<any text>".

"<any text>" is expressed exactly as it is written in this ini file.

The following configuration possibilities exist for the area or line index:

<CH>	for the channel number in which the axis is active.
<AX>	for axis number
<DRV>	for the drive number of the associated axis
<PLC_CH>	for a PLC-channel DB; is replaced by 20+ channel number of the channel in which the channel is active.
<PLC_AX>	for a PLC axis DB; is replaced by 30+axis number; e.g.: Item = /Channel/MachineAxis/impulseEnable[u<CH>, <AX>] (Item: (see description of the OPI).

In this case the current channel would be used for <CH> and the current axis number for <AX>. The value obtained can be evaluated by bits or as a whole number.

expr: This can be an IF statement or a bitmap file assignment.

Syntax:

```
expr=<IF_Statement> or
expr=<Bmp_Assignment>
<IF_Statement>::= IF <BoolExpression> THEN
<Statement>ELSE<Statement>
<Bmp_Assignment>::= BMP = <BitmapNr>
<Statement>::= <EntryName> or
<Statement>::= <Bmp_Assignment>
<BoolExpression>::= <Val> or
<BoolExpression>::= <Val><OP><Value>
<Val>::= # or
<Val>::= #.<BitNr>
<BitNr>::=Bit number to be evaluated (0..31)
<OP>::= < , > ,= ,>=, <=, <>
```

An expression must occupy one line.

<EntryName>::= any name, which must be defined in the same section as expr. It is treated in the same way as expr.

<BitmapNr>::= Number of a bitmap file. The numbers 0 to 99 are reserved for Siemens. Internal bitmaps must be defined in the [BMP] section.

The predefined numbers have the following significance:

0:		do not display bitmap	
1:	OK	🟢	Displaying the bitmap
2:	Not OK	🟡	Displaying the bitmap
3:	Error → alarm is present	🔴	Displaying the bitmap
4:	Not applicable	●	Displaying the bitmap
5:	No drive allocated	–	Displaying the bitmap
6:	Communication error	#	Displaying the bitmap

```
[BMP]
```

Any customized bitmaps can be specified in this section. The first bitmap in the [BMP] section must have number 100, the second 101 ...

The name must be specified in double inverted commas. If no path is stated, the system searches for the bitmap in mmc2 or addon, oem, user…, otherwise in the specified path.

Example: Signal for measuring system 1 active

```
[GLOBAL]
NrOfSignals=1
[BMP]
100="test.bmp"
101="c:\tmp\test2.bmp"
```

```
[SIG1]
text= "Measuring system 1 active"
Item= /Nck/MachineAxis/encChoice[u1, <AX>]
expr= if #=1 then BMP=100 else expr2
expr2= if #=0 then BMP=101 else BMP=2
```

Structure of the DGOVWTXT_XX.INI File

xx in the file name stands for the language abbreviation for the language-specific file.

The only section is the [TEXT] section.

The individual tests are listed here in the form:

$T<Text No.>**= "<**any text>**"** listed.

Values from 1000 to 32767 are permitted for <Text-Nr> and may only occur once. Values below 1000 are reserved for Siemens.

This file has to exist for all languages that are required (language abbreviation as in MMC.INI).

See also

DGOVW.INI (Page 279)

2.1.10 Setting the Wide Display for OEM applications

Using the Wide Display

The HMI program is displayed on a large screen as "Wide Display" in such a way that an area for OEM applications remains free in the upper section.

Upper Screen

The area above the output range of the HMI program is identified as the Upper Screen. This area has a size of 1280 x 254 pixels.

Requirements

To activate the Wide Display function, the following requirements apply:

- Wide Display will only work if the the screen has a resolution of at least 1280 x 1024 pixels and the "UpperScreen" function in the REGIE.INI configuration file, section [UpperScreen] is set to TRUE.

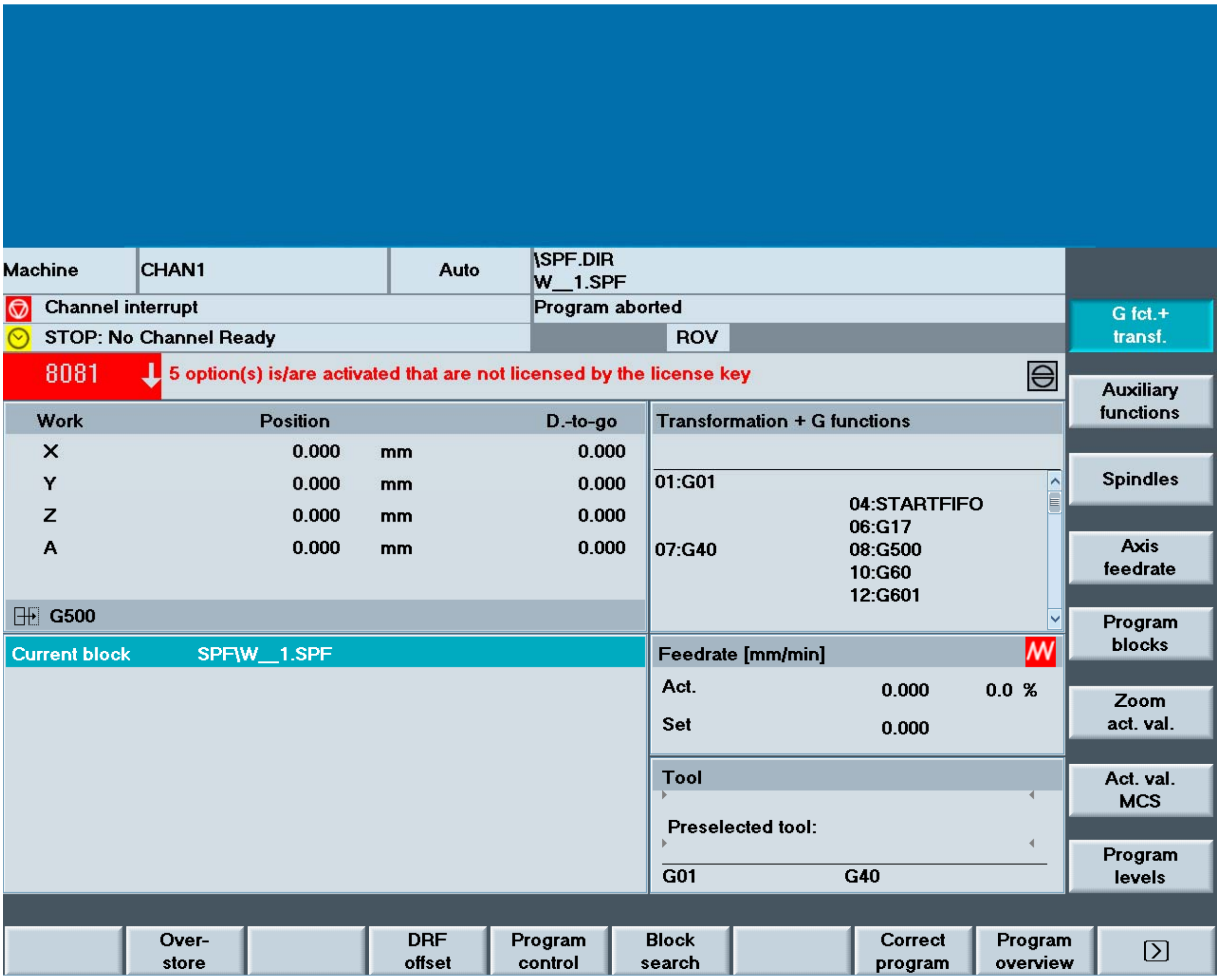
  Procedural control stores the start position of the HMI program in the HMI_Start entry in section [CONTROL] in the MMC.INI file.
- OEM applications that wish to use Wide Display must be regenerated with the current OEM package and the expansion of procedural control.

Activating the function

REGIE.INI configuration file section:

```
 [UpperScreen]
; Upper screen area above HMI-Wide-Screen.
; The upper screen area is managed by an independent native windows
; application independently of the REGIE management.
; The upper screen area has an independent softkey area.
; This upper screen softkey area is available only
; on special OEM operator panels.
; Activate this function here
UpperScreen = TRUE
; Announce main window of upper screen,
; to transfer the softkeys from Regie.
; WindowName
UpperScreenWindowName = ""
; Window ClassName
UpperScreenClassName = ""
```

Example



Machine | CHAN1 | Auto | \SPF.DIR W__1.SPF

Channel interrupt | Program aborted

STOP: No Channel Ready | ROV

8081 5 option(s) is/are activated that are not licensed by the license key

Work	Position		D.-to-go
X	0.000	mm	0.000
Y	0.000	mm	0.000
Z	0.000	mm	0.000
A	0.000	mm	0.000

G500

Current block SPF\W__1.SPF

Transformation + G functions

01:G01
04:STARTFIFO
06:G17
07:G40
08:G500
10:G60
12:G601

Feedrate [mm/min]		
Act.	0.000	0.0 %
Set	0.000	

Tool

Preselected tool:

G01 G40

Figure 2-2 Example HMI-Advanced

Operating the HMI program and OEM application

NEXT WINDOW

When switching the operation between HMI program and OEM application, use the window shift key <NEXT WINDOW> or simply click.

The OEM application is operated in the main window, either via actuation or by a mouse click. All other keyboard entries will now appear in this active window until one of the function keys (F1-F10, etc.) recognized by the HMI program is detected. The HMI program is then active again.

The following shortcuts are reserved:

<Ctrl+1>, <Ctrl+2>, <Ctrl+3>, <Ctrl+6>, <Ctrl+7>, <Ctrl+8>

2.1.11 Creating technology-specific texts

Function

The following sections are available in the IF.INI file for parameterizing the Help functions:

- `TECHNOLOGY`
- `TEXTFILES`
- `CONTEXT`
- `TEXTSEARCH`

Section [TECHNOLOGY]

The technology group to which the machine belongs can be specified in the [TECHNOLOGY] section in order to limit the display of instruction texts to those entries that are relevant for the machine. Then, only entries that are relevant to the technology group are displayed.

The following technology codes can be specified:

"m" = Milling

"t" = Turning

"g" = Grinding

"s" = Nibbling

"p" = Punching

"a" = Display all technologies

Section [TEXTFILES]

The paths (path\name) of the Quick Help text files on which the help function is based are parameterized in the [TEXTFILES] section.

The help function accesses the following text files:

- Siemens dh\cst.dir\ifs_gr.com
- Machine manufacturer dh\cst.dir\ifm_gr.com (any file name)
- End user dh\cst.dir\user_gr.com (any file name; is also entered under "Settings")

Note

The path for the Siemens text file "Dh\Cst.dir\Ifs_gr.com" must exist; if not, an error message is output accordingly.

The entry for Siemens is specified in conjunction with ISO language as:

SiemensI=file

where file equals ifit_xx.com for turning or

where file equals ifim_xx.com for milling.

There are different parameterization options for the names of the text files when assigning the text path.

The following table lists the parameterization options and their interpretation by the Help function:

Parameters for the text file	Interpretation
FileName.com	The text file that is independent of the language is read-out
File name_XX.com	The language-dependent text file is read out
FileName_.com	In this setting the non-language-specific test file is read first, if present, otherwise the language-specific text is read.

(XX = abbreviation for languages, e.g. "GR" for German)

Example: Enduser=Cus.dir\Ifm_.com

First of all, the non-language-specific text file "Ifm.com" is opened in the path "C:\Dh\Cus.dir\" by default or, if the non-language-specific text file does not exist, the relevant language-specific text file (e.g. "Ifm_gr.com") is opened.

Extract from Short Help txt file "Ifm_gr.com":

```
...
a2//G00/Linear interpolation with rapid traverse (m)
a2//G01/Linear interpolation with feed (m)
a2//G02/Circular interpolation in the clockwise direction (m)
...
```

Section [CONTEXT]

An increased context sensitivity of the Help function can be selected in the [CONTEXT] section. This setting is activated with "1" and deactivated with "0" (is also entered under "Settings").

If the increased context sensitivity is activated and the cursor is positioned to the left of, next to or on an instruction text, then all instructions with the same initial letters are displayed when Help is called.

For example, if the programmer has selected instruction "G1", then all instructions beginning with "G1" (e.g. "G1", "G17", "G18", "G19") are displayed.

If increased context sensitivity has been deactivated with "0", then only the instruction actually selected (if available) is displayed.

Section [TEXTSEARCH]

The text search type can be specified in the [TEXTSEARCH] section.

The following search methods can be parameterized:

1 = instruction texts only

2 = descriptive texts only

3 = instruction and descriptive texts (are also entered via "Search").

See also

IF.INI (Page 290)

2.1.12 Set-up workpieces with job lists

Requirements

The option "multiple channel sequential programming" and a multiple channel machine with 20 channels are prerequisites. An offline mode is possible; the channels cannot be checked in this case.

Function

The templates for work pieces offer a simple possibility of creating new program sequences in the form of workpieces with job lists. These templates can be adapted for an existing workpiece.

In the program operating area, a workpiece is processed using "Edit workpiece." If no joblist, part program or DAT file with the workpiece name exists, an error message appears stating that the workpiece cannot be opened. If an automatic adaptation of the workpieces is activated by an entry in the MMC.INI file, an attempt is made to finish the workpiece instead of an error message.

Parameterizing

The following parameters must be specified in the [Program] section in the MMC.INI file:

```
 [Program]
; 1=finished workpiece via workpiece template,
; if cannot be opened with MCSP (default)
; 0=outputs an error message:
; UpdateWorkpiecesBasedOnTemplates=1
```

There is only one workpiece template under templates with JOB/DAT file. The contents of this template are then automatically copied into the selected workpiece. Files that already exist in the current workpiece are not overwritten. If a JOB/DAT file with the workpiece name exists after this, the sequence editor is opened, otherwise an error message appears.

If there are several workpiece templates under templates with JOB/DAT files, a "Finish workpiece" dialog appears which allows the desired workpiece template to be selected.

If a part program that does not contain an INIT instruction is selected when "Edit workpiece" is activated, the "Finish workpiece" dialog is called up, which allows a selection to be made from the JOB/DAT files (if available) under templates. If a JOB/DAT file is selected, it is copied into the workpiece under the part program name and opened with this file of the sequence editor. Workpiece templates are not offered at this point, but only when a workpiece is selected.

The files and messages stating that the template files are already contained in the workpiece are displayed in the log window.

Error messages

Error messages are displayed in the following cases:

If a part program that does not contain a valid INIT instruction is selected:

```
No INIT instruction found in the part program!
```

There is a reference to a non-existent file in the JOB/DAT file:

```
The following non-existent programs are addressed:
```

There is a reference to a non-existent channel number in the JOB/DAT file:

```
Programs that do not exist
are assigned to the following channels:
```

If no entry could be created in the DAT file:

```
No channel assignment found!
```

Additional configuring possibilities for multi-channel step sequence programming:

See also

MMC.INI (Page 303)

SEDITOR.INI (Page 317)

2.1.13 Tool management (WZV)

Function

The tool management HMI-Advanced already contains this functionality.

The structure of the NCDDE variables of tool management follows the pattern:

```
"TMHMICurData_"<MMCName>"_"<DataName>
```

with TMHMI = Tool Management Human Machine Interface

The following current data (<DataName>) are present:

General TOA and channel data:

```
- curTOANo current TOA number
- curChannelNo current channel number
```

Current tool in list views (tool in which the cursor lies) and in tool data images for individual tools:

```
- curToolTNo T number
- curToolIdent tool identification
- curToolDuplo tool duplo number
- curToolType tool type
- curEdgeNo tool edge number, relative to the tool,
  not the the Duplo number !
- curDLNo tool edge correction location number,
 relative to the edge
```

```
- curMagNo Magazine number; "0", if the current tool
 is not to be found in or intended for a tool location.
- curPlaceNo magazine location number; "0" if current tool
 is not to be found in or intended for a tool location.
```

Current magazine in magazine list view:

```
- curMagLiMagNo magazine number,
```

Target magazine, target magazine place during loading, unloading, reloading, positioning, empty place search:

```
- targetMagNo magazine number
- targetPlaceNo magazine place number
```

Source magazine, source magazine place during loading, unloading, reloading, positioning, empty place search:

```
- sourceMagNo magazine number
- sourcePlaceNo magazine place number
```

Current tool in tool cabinet:

```
- curCabToolIdent tool identification
- curCabToolDuplo tool duplo number
- curCabToolType tool type
Current tool in tool catalog:
- curCatToolIdent tool identification
- curCatToolDuplo tool duplo number
- curCatToolType tool type
```

Additional information needed for commissioning can be found in:

References: Function Manual Tool Management

2.1.14 Tool selection without tool management

Function

The "Parameters without tool management" application provides local NCDDE variables that write to the currently selected tool. The tool management functions are used for this purpose for tool selection under Parameters. The reason why these variables are written is influenced by the settings in PARAM.INI: Variables are written to only when images change as a result of "Expand user interface" or, alternatively, at each parameter status change.

The tool is selected with the cursor in the tool images without tool management. When the cursor is positioned on a tool, the internal T number and the edge number are written to local NCDDE variables for this tool. These variables can be evaluated from "Expand user interface."

All data in one variable

The name of the NCDDE variables follows the pattern:

```
"PAHMICurData"_< MMCName>, with MMCName from MMC.INI.
```

Thus, the difference between the name of the NCDDE variables and the tool management variables lies in the prefix.

Note

The NcddeMmcName and NcddeMmcName entries in the [GLOBAL] section must not have standard value "__xxxx__". Otherwise, it will be replaced by a random number.

In this variable, the current PAHMI data are fed as a string, possible in the form:

"curToolTNo=35;curToolDNo=5;" with "=" as a separator between data name and data value and "," as a separator between data

Individual variables

In addition (or alternatively), a separate NCDDE variable is available for data element. The name of this variable is formed from the following:

```
"PAHMICurData_"<MMCName>"_"<DataName>
```

where <MMCName> is taken from the mmc.ini file in the [GLOBAL] section in the NcddeMmcName entry.

Note

The NCDDE server does not permit multiple variable access to NCDDE variables so that read and write accesses occur individually.

The following current data (<DataName>) for TOA and channel are present:

```
- curTOANo Current TOA number
- curChannelNo Current channel number
```

Current tool in list views (tool in which the cursor lies) and in tool data images for individual tools:

```
- curToolTNo T number
- curToolType Tool type
- curEdgeNo Tool edge number, relative to the tool, not DNo!
```

Currently unknown values exist as "varname=;"" in the NCDDE variables or are omitted. The data sequence is not defined. As long as a data value is not set, its NCDDE variable is blank or is not present.

Control of interface

The interface activity can be controlled using PARAM.INI.

```
[General]
; Options of "write current data of tool management human
; interface to ncdde variables when switch to
: WIZARD forms or WIZARD softkeys occur":
; All Options have to be set by using named arguments in
; one single line.
; Options are
; switched "ON" by value "True" and
; switched "OFF" by value "False" or if named argument
; does not exist in line.
; "EnableAllTogetherWriteToNcdde := True":
; all data in one ncdde variable
; "EnableSingleWriteToNcdde := True":
; one ncdde variable per data
; Both options can be active.
; If both options are not active,
; no write to ncdde variables will occur.
; "WriteChangesWhenStateChanged := True":
; write changes when a softkey is pressed,
; even if there is no switch to WIZARD
```

The following setting must be in a single line of PARAM.INI:

```
HMICurDataInterface = EnableAllTogetherWriteToNcdde := True,
EnableSingleWriteToNcdde := True,
WriteChangesWhenStateChanged := False
```

2.1.15 Executing from the hard disk (m:n configuration)

Requirements

This function applies to SINUMERIK powerline only.

Included in the NETNAMES.INI file are descriptions of the following:

- Configuration of assignments of multiple HMI components and multiple NCU components (m:n configuration).

  If, for example, multiple HMI components have been assigned to one NCU, the unit that is to provide the part program in the event of "Execution from external source (hard disk)" must be specified.
- Configuring the channel menu for multi-channel systems
- Reference to file with symbols for PLC signals

Function EXTCALL

In order to execute external programs that are located on the hard disk of the PCU in the data management system from the part program using the EXTCALL program command, the following entries, for example, are required in the section [conn HMI_x] of the NETNAMES.INI file.

```
; HMI identification part
[own]
owner= HMI_3
; Description of possible connections
; Connection part
[conn HMI_3]
```

```
conn_1= NCU_1
conn_2= NCU_2
conn_3= NCU_3
conn_4= NCU_4
EXTCALL_conns=conn_1, conn_2, conn_3, conn_4
; conn_i identifier only for the NCUs from which the PCU
; can be accessed (with HMI) (see note)
; Description of significant net parameters
[param network]
bus= opi
; HMI descriptions
[param HMI_3]
HMI_typ = 0x40
HMI_......

; descriptive part, NCU components
[param NCU_1]
type =NCU_573
nck_address= 11
plc_address= 11
name = NCU1
...

; Reference to the file with symbols for PLC signals from NCU_1
PlcSymbolFile=PlcSym
```

Note

The entry EXTCALL_Conns … is only needed in the "main operating field" in the NETNAMES.INI file, not in the "secondary operating fields".

See also

NETNAMES.INI (Page 308)

References: Function Manual, Expansion Functions: Several operator panel fronts on several NCUs, distributed systems (B3)

2.1.16 Activating the V.24 interface

Activating the V.24 interface

The serial V.24 interface can be activated in the DINO.INI file, section [System]. Here it is possible to save files and assign them to the V.24/PG (COM1/COM2) interfaces.

The default setting is 0 (no interface parameterization).

```
[System]
; enable V24 setting by setting a non zero value
V24Settings=0
```

Procedure

To be able to use the V.24 data transfer with HMI-Advanced 7.5 on PCU 50.3, the following steps are required:

1. In the BIOS of the PCU 50.3 under "Advanced → I/O Devices Configuration", set the "Internal COM1" to "Disabled".
2. Insert the "PCI Interface Card (COM/LPT)" into the PCU 50.3 and install the supplied driver.
3. In the Windows device manager, rename the COM interfaces as COM1, COM2.
4. In the Windows device manager, for each COM interface, under "Port Settings → Advanced", deactivate the "Enable CTS/RTS Auto Flow Control" option.
5. Activate the V.24 interface in the DINO.INI configuration file.

See also

DINO.INI (Page 277)

2.2 Configuring a channel menu

2.2.1 Applications for the channel menu

Applications

1 ... n
CHANNEL

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

It is configured in the NETNAMES.INI file. The "control unit management" option is required for operation station switchover for SINUMERIK powerline. That means maximum 9 PCUs can be operated on 9 NCUs for the m:n configuration or 1 PCU can be operated on max. 20 NCUs (powerline).

The channel menu can be used for the following applications:

- Without control unit management (1:1 configuration) for switchover of the NCK channels for a larger number of NC channels for direct channel selection (local channel switchover). Benefit is direct channel selection instead of sequential switching through the channels with the channel switchover key until the channel is found.
- Without control unit management (1:1 configuration) for display switchover in double-channel display (no channel menu displayed!)
- With control unit management (M:N configuration) for switching the control unit (OP and MCP) to another NC channel, if necessary, with implicit switchover to another NCK. The entire HMI system is switched over. This is the mode for control unit management for SINUMERIK powerline.
- With control unit management for switchover of the operating station with TCU to another HMI (and therefore to another NCK) or to another channel of the displayed NCK while keeping the displayed HMI (T:M:1 scenarios). This is the mode for control unit management for SINUMERIK solution line.

For clarification, a graphical representation of the differences for SINUMERIK powerline and solution line. A local channel switchover (while keeping the NCK connected) is not shown:

Switchover procedure for SINUMERIK powerline

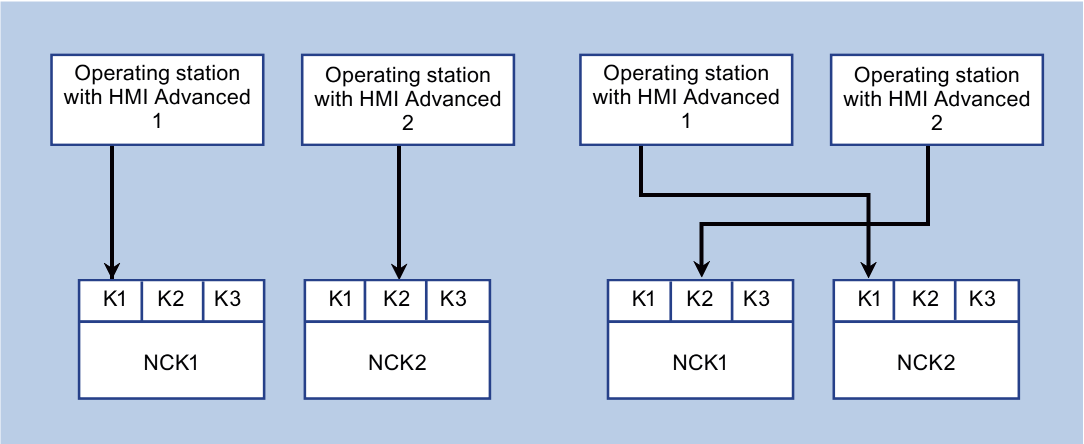

Operating station with HMI Advanced 1
Operating station with HMI Advanced 2
Operating station with HMI Advanced 1
Operating station with HMI Advanced 2
K1
K2
K3
NCK1
K1
K2
K3
NCK2
K1
K2
K3
NCK1
K1
K2
K3
NCK2

Figure 2-3 Switchover (powerline)

Initial situation:	After switchover:
HMI Advanced 1 to NCK1.channel1	HMI Advanced 1 to NCK2.channel1
HMI Advanced 2 to NCK2.channel2	HMI Advanced 2 to NCK1.channel2

Switchover procedure for SINUMERIK solution line

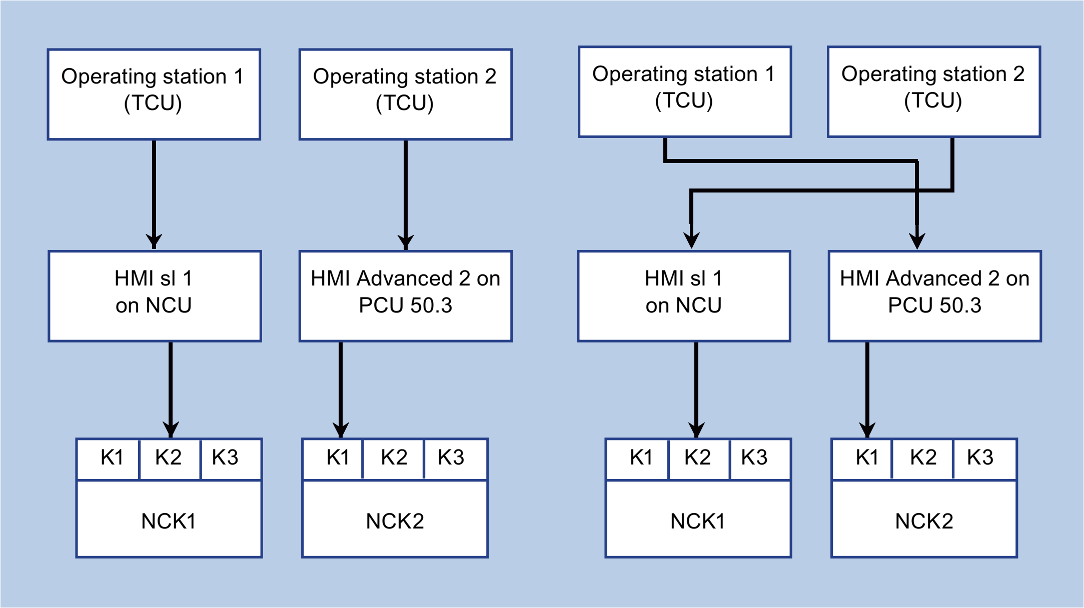

Operating station 1 (TCU)
Operating station 2 (TCU)
Operating station 1 (TCU)
Operating station 2 (TCU)
HMI sl 1 on NCU
HMI Advanced 2 on PCU 50.3
HMI sl 1 on NCU
HMI Advanced 2 on PCU 50.3
K1
K2
K3
NCK1
K1
K2
K3
NCK2
K1
K2
K3
NCK1
K1
K2
K3
NCK2

Figure 2-4 Switchover (solution line)

Initial situation:	After switchover:
Operating_station1 (on HMI1) to NCK1.channel1	Operating_station1 (on HMI2) to NCK2.channel1
Operating_station2 (on HMI2) to NCK2.channel2	Operating_station2 (on HMI1) to NCK1.channel2

2.2.2 Structure of the channel menu

Overview

With the exception of the application cases for the double-channel display, a channel group list is defined for switching over operation for the channel menu.

A channel group list comprises 1 or several channel groups. A channel group list comprises 1 or several NC channels. In the channel menu, the channel groups can be selected via the horizontal softkeys The channels of a selected channel group can be selected with the vertical softkeys. A possible switchover target is therefore always the NC channel of a real NCK. Therefore switchover to another channel triggered via the channel menu can implicitly mean switchover to another NCK.

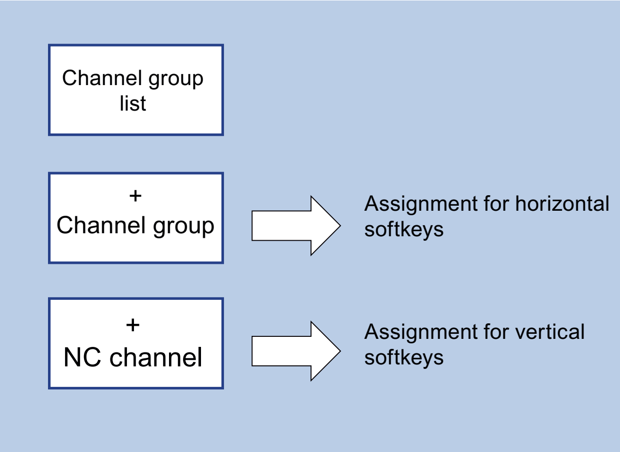

Channel group list
+
Channel group
Assignment for horizontal softkeys
+
NC channel
Assignment for vertical softkeys

Figure 2-5 Assignment of softkeys

Note

In a system, for example, an operating station can correspond to a channel group.

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

2.2.3 Configuration of a channel menu for direct channel selection (1:1 configuration)

Configuring a channel menu

You configure the channel menu in the NETNAMES.INI configuration file, see example below: 4 channel groups with differing numbers of channels are configured on an NCU.

```
;********************************************* **********
; ChanMenu
;*********************************************** ********
[chan MMC_1]
DEFAULT_logChanSet = Stat_1
DEFAULT_logChan = CH_1
ShowChanMenu = true
logChanSetList = Stat_1, Stat_2, Stat_3, Stat_all

;********************************************************
; definition of the channel groups

[Stat_1]
logChanList = CH_1, CH_2, CH_3, CH_4
```

```
[Stat_2]
logChanList = CH_5, CH_7, CH_8

[Stat_3]
logChanList = CH_6

[Stat_all]
logChanList = CH_1, CH_2, CH_3, CH_4, CH_5, CH_6, CH_7, CH_8

;**********************************************************
; definition of the channels

[CH_1]
logNCName = NCU_1
ChanNum = 1

[CH_2]
logNCName = NCU_1
ChanNum = 2

[CH_3]
logNCName = NCU_1
ChanNum = 3

[CH_4]
logNCName = NCU_1
ChanNum = 4

[CH_5]
logNCName = NCU_1
ChanNum = 5

[CH_6]
logNCName = NCU_1
ChanNum = 6

[CH_7]
logNCName = NCU_1
ChanNum = 7

[CH_8]
logNCName = NCU_1
ChanNum = 8
```

Switching over channels

1 ... n
↓ ↓
CHANNEL

The following channel menu is displayed when the channel switchover key is pressed with this configuration.

Select the channel menu using the channel switchover key from any operating area of the HMI, as shown in the following example:

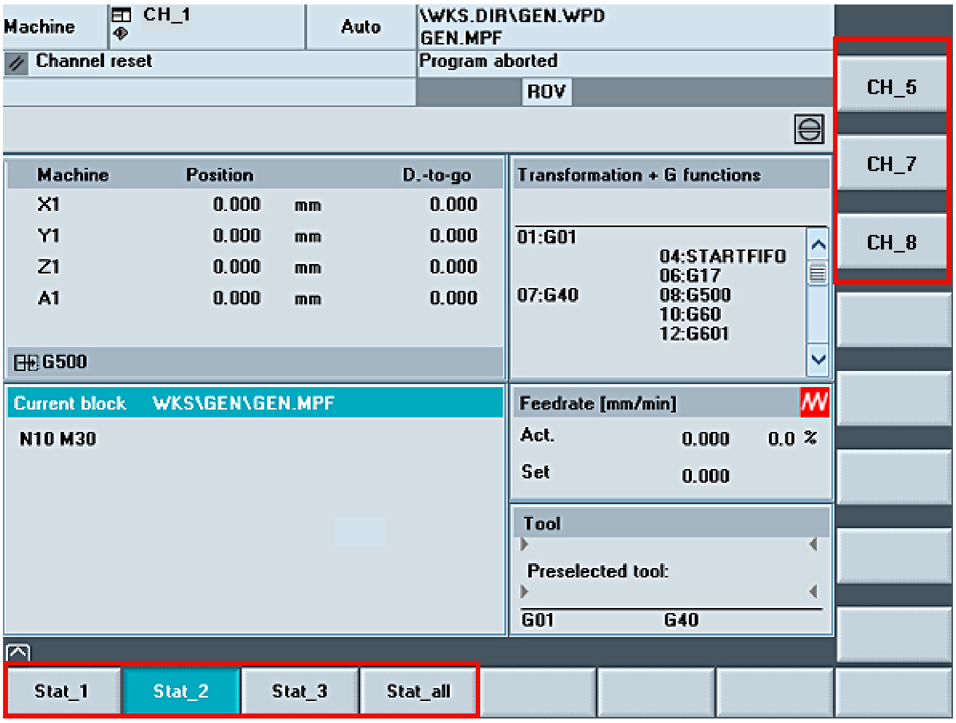


Figure 2-6 Example channel menu

Triggering channel switchover:

- Select the channel group (station) with the horizontal softkeys.
- The channels belonging to this group are now displayed on the vertical softkeys.

  The control does not switch to the channel until the vertical softkey is pressed.

See also

NETNAMES.INI (Page 308)

2.2.4 Configuring a double-channel display

Configuring a double-channel display

If a double-channel display is used, the control unit management options cannot be used. Before the double-channel display in the "Machine" operating area, the following entries must be made in file NETNAMES.INI:

```
[own]
owner= HMI_1

[chan HMI_1]
ShowChanMenu=false

[MULTI_CHAN_i]
CHAN_2 = k
```

" i " denotes the channel number in the current NCU. The second channel now displayed is specified in entry CHAN_2: CHAN_2 = k

This default setting of the configuration only applies when switching from 1-channel to 2-channel display. If 2-channel display is active and a channel is switched within the display, the other channel remains unaffected.

Note

Either double-channel display or control unit management can be used

For double-channel display, ShowChanMenu=false must be set.

Each of the channels involved in the 2-channel display appear once in [MULTI_CHAN_i] and once as 2nd channel in CHAN_2=i.

For double-channel display, display MD 9034: MA_NUM_DISPLAYED_CHANNELS must be set to 2.

Example of a double-channel display

On an NCU with 6 channels (1, 2, 3, 4, 5, 6), channel combinations 1 and 6, 2 and 5, and 3 and 4 must be displayed together:

```
[MULTI_CHAN_1]
CHAN_2 = 6

[MULTI_CHAN_6]
CHAN_2 = 1

[MULTI_CHAN_2]
CHAN_2 = 5
```

```
[MULTI_CHAN_5]
CHAN_2 = 2

[MULTI_CHAN_3]
CHAN_2 = 4

[MULTI_CHAN_4]
CHAN_2 = 3
```

2.2.5 Channel menu for control unit management for an m:n configuration (powerline)

m:n configuration

m:n configuration means a group comprising m HMI Advanced units and n NCU units. The HMI units for operating the relevant NCUs can be freely switched over during operation. For each of the switchover targets of an HMI, one channel menu is configured in the NETNAMES.INI file for each individual HMI. The individual switchover targets are declared as a pair comprising an NCU and the processing channel on that NCU. Because the machine control panel (MCP) for an HMI also has to be activated when that HMI is switched to another NCU, the machine control panel must be assigned to the HMI in the configuration.

The configured channel menu is called up on the HMI by pressing the channel switchover key. The switchover target can be selected and switchover triggered via the softkeys.

NOTICE
One NCU may only have two active connections (HMIs) at any one time. This must be taken into account for all HMIs in the configuration in NETNAMES.INI for the entries DEFAULT_logChanSet and DEFAULT_logChan with which the connection to be established is defined after booting.

The following switchover options are described by the configuration specified in the example of the NETNAMES.INI in the appendix:

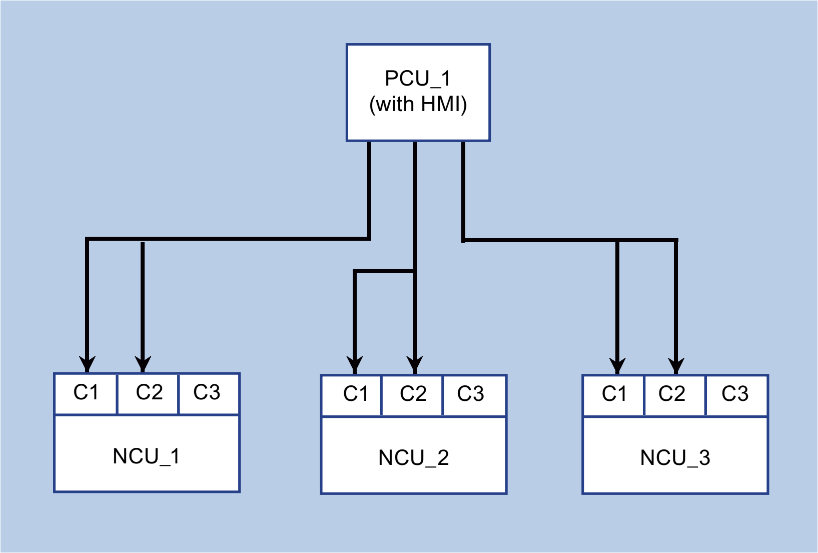


Figure 2-7 Switchover options

See also

- On configuring the channel menu for the application case: NETNAMES.INI (Page 308)
- On displaying the channel menu in HMI Advanced: Configuration of a channel menu for direct channel selection (1:1 configuration) (Page 53)

2.2.6 Channel menu for control unit management for an m:n configuration (solution line)

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 of each operating station are individually configured in a channel menu.

The HMI systems remain permanently linked to the NCK they are assigned to. It is irrelevant whether the HMI is an embedded HMI that runs on the NCU or a so-called HMI that runs on a PCU. However, only one HMI may be linked to each NCK (statically), i.e. if an external HMI Advanced is to operate the NCK of an NCU, HMI sl must be deactivated there (see: Service command sc disable HMI).

During operation, switchover is initiated via the configured channel menu from the HMI.

From the point of view of the HMI, t:m:n operation is active if a channel menu is configured in the NETNAMES.INI file.

Available switchover targets are defined via the address of the HMI and optionally to an NC channel that must be assumed on switchover from the HMI. These switchover targets can be combined in 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.

Two-stage procedure

In the first stage, all channel groups that are to be used in the system are defined in file NETNAMES.INI, irrespective of the actual operating stations.

A channel group is specified as follows:

```
[CH_GRP_1]                        /*  Name of the channel group; the notation
                                  /*  must follow this scheme.
                                  /*  The groups are consecutively numbered
                                  /*  in the last digit.

Text = Process1                   /*  Softkey text for the channel group for
                                  /*  display in the channel menu (horizontal
                                      softkey)

Member1.Target = NCU730PN-IP3.#1  /*  Switchover target 1 of channel group
                                  /*  computer name (host name) or
                                  /*  permanent IP address of the computer
                                  /*  on which the HMI to be selected runs
                                  /*  (NCU or PCU) ".#channelnumber" can
                                  /*  be optionally used to specify the NC
                                  /*  channel to be assumed on switchover.
                                  /*

Member1.Text = NCU730_K1          /*  Softkey text for display in the channel
                                  /*  menu (vertical softkey) Texts containing
                                  /*  a blank must be quoted in " ".

Member2.Target =                  /*  analog
Member2.Text =                    /*  analog
```

The file NETNAMES.INI must be distributed to all systems (NCU, PCU) on which an HMI runs, i.e. for HMI sl on NCU, this file must be placed in directory /oem/sinumerik/hmi/cfg or /user/sinumerik/hmi/cfg, for HMI Advanced it must be placed in directory F:\OEM or F:\USER\.

The second step is to define the characteristics of each individual operating station by specifying the channel groups relevant to that station in the station's own CONFIG.INI file in section [T2M2N].

The CONFIG.INI files are located on the NCU in the directories:

```
/user/common/tcu/<TCU-NAME>/common/tcu/config.ini
```

On the PCU 50.3, they are to be found at:

```
F:\user_base\common\tcu\<TCU-NAME>\common/tcu/config.ini
```

For <TCU-NAME>, the TCU name of the operating station, as assigned on the system during initial booting of a TCU, must be used.

Differences between HMI sl and HMI Advanced

For HMI Advanced, a channel menu for the direct channel selection must be configured in addition to the channel groups that are used for switching over the operating stations. This configuration is not evaluated or might be missing for HMI sl.

The HMI Advanced then responds as follows:

- If the PCU 50.3 has a local display and if this local display has the operating rights, channels are selected directly via the channel menu.
- If an operating station with TCU for which no channel menu is configured in its file CONFIG.INI has the operating rights on PCU 50.3, channels are selected directly via the channel menu.
- If an operating station with TCU for which one channel menu is configured in its file CONFIG.INI has the operating rights on PCU 50.3, operating station switchover is performed via the channel menu as defined in the configuration.

HMI sl responds as follows:

- If an operating station for which no channel menu is configured has the operating rights, the channel switchover key is local, without a channel menu, i.e. switching to the next NC channel.
- If an operating station for which one channel menu is configured in its file CONFIG.INI has the operating rights, operating station switchover is performed via the channel menu as defined in the configuration.

See also

On displaying the channel menu in HMI Advanced: Configuration of a channel menu for direct channel selection (1:1 configuration) (Page 53)

Configuration of the channel groups (NETNAMES.INI)

```
[own]
owner= HMI_1

[chan HMI_1]                    /* channel menu for direct channel
selection
DEFAULT_logChanSet=channels      /* obligatory for HMI Advanced
DEFAULT_logChan=K_1             /* of no significance for HMI sl
ShowChanMenu=True
logChanSetList=channels

[channels]
logChanList=K_1, K_2, K_3, K_4

[K_1]
logNCName=NCU_1
ChanNum=1
```

```
[K_2]
logNCName=NCU_1
ChanNum=2

[K_3]
logNCName=NCU_1
ChanNum=3

[K_4]
logNCName=NCU_1
ChanNum=4

;*************************** /* channel menu for TCUs
[chan HMI_1]                 /* common part for HMI Advanced
ShowChanMenu = true          /* and HMI sl


;Channel groups
;Correlation from channel group to horizontal softkey
;-> see config.ini of the TCU

[CH_GRP_1]

Text = Process1

Member1.Target = NCU730.#1
Member1.Text = NCU730_K1

Member2.Target = NCU730.#3
Member2.Text = NCU730_K3

Member3.Target = NCU730.#4
Member3.Text = NCU730_K4

[CH_GRP_2]

Text = Process2

Member1.Target = NCU710-2.#1
Member1.Text = NCU710_K1

Member2.Target = NCU710-2.#2
Member2.Text = NCU730_K3
```

```
[CH_GRP_3]

Text = Transport

Member1.Target = NCU720-1.#1
Member1.Text = NCU720_K1

Member2.Target = NCU720-1.#2
Member2.Text = NCU720_K2

Member3.Target = NCU720-1.#3
Member3.Text = NCU720_K3

Member4.Target = NCU720-1.#4
Member4.Text = NCU720_K4

…

[CH_GRP_15]

Text = Headend

Member1.Target = 192.168.214.241.#1
Member1.Text = NCU720_K1

Member2.Target = 192.168.214.241.#2
Member2.Text = NCU720_K2

Member3.Target = 192.168.214.241.#3
Member3.Text = NCU720_K3

Member4.Target = 192.168.214.241.#4
Member4.Text = NCU720_K4
```

Configuration for an operating station

The channel menu for an operating station is configured in its file CONFIG.INI in section [T2M2N]. There the channel groups to be used for the operating station as defined in file NETNAMES.INI are assigned to the horizontal softkeys of the channel menu for this operating station.

File CONFIG.INI

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

The references CH_GRP_xx of the softkeys refer to the channel groups which must be defined in NETNAMES.INI. This notation is binding.

This example and the conditional configuration in NETNAMES.INI defines that the channel menu for this operating station must be assigned as follows:

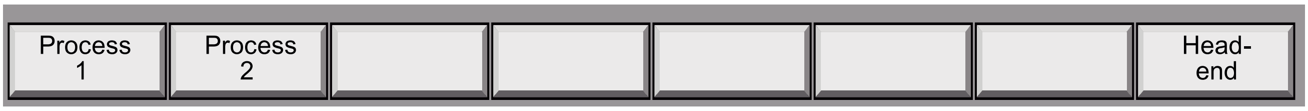


Figure 2-8 Horizontal softkey menu

2.2.7 Management of operating units 1:N (solution line)

Activation of the function

Configuration file mmc.ini:

```
[Global]
NcddeMachineName = MachineSwitch
NcddeDefaultMachineName = net:NCU_1
NcddeMachineNames = net,NCU840D
```

Examples of netnames.ini for 1 HMI to 3 NCUs:

```
[own]
owner= HMI_1

; Description of possible connections
[conn HMI_1]
conn_1= NCU_1
conn_2= NCU_2
conn_3= NCU_3

[param HMI_1]
mmc_address = 1
```

```
[param NCU_1]
nck_address=
192.168.214.1,LINE=10,NAME=/NC,SAP=030d,PROFILE=CLT1__CP_L4_INT
plc_address=
192.168.214.1,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L4_INT
name=Machine_1

[param NCU_2]
nck_address=
192.168.214.2,LINE=10,NAME=/NC,SAP=030d,PROFILE=CLT1__CP_L4_INT
plc_address=
192.168.214.2,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L4_INT
name=Machine_2

[param NCU_3]
nck_address=
192.168.214.3,LINE=10,NAME=/NC,SAP=030d,PROFILE=CLT1__CP_L4_INT
plc_address=
192.168.214.3,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L4_INT
name=Machine_3

;***************************
; ChanMenu
;***************************
[chan HMI_1]
DEFAULT_logChanSet = Machine_1
DEFAULT_logChan = NCK1.1
ShowChanMenu = true
logChanSetList = Machine_1, Machine_2, Machine_3

[Machine_1]
logChanList = NCK1.1

[NCK1.1]
logNCName = NCU_1
ChanNum = 1

[Machine_2]
logChanList = NCK2.1

[NCK2.1]
logNCName = NCU_2
ChanNum = 1

[Machine_3]
logChanList = NCK3.1
```

```
[NCK3.1]
logNCName = NCU_3
ChanNum = 1
```

Note

Function 1:N for solution line

- An HMI-Advanced can be switched over to as many as 4 NCUs.
- Drive alarms from an NCU which is not currently linked with the HMI are displayed as a group alarm. After the HMI has been switched over to the relevant NCU, the drive alarms are shown in resolved form with all the details displayed.
- The PLC block FB9 must not be used when switching over an MCP with the HMI (FB9 must not be active). If necessary, the MCP must be switched over in the PLC user program.

Solution

In the OB100 (the 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. MCP is now active.

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

See also:

Function Manual Expansion Functions: Several Operator Panels on Several NCUs, Distributed Systems (B3)

2.3 Connecting network drives

2.3.1 Setting up access to external drives or computers

Use

The control can be connected to external network drives or other computers. This makes it possible, for example, to execute part programs from other computers.

Two alternatives are available:

- A maximum of 4 drives can be connected by parameterizing display machine data.
- A maximum of 8 drives can be connected by configuring the LOGDRIVE.INI file.

The two options cannot be applied jointly. The LOGDRIVE.INI file is the preferred choice for applications with multiple TCUs or PCUs because it allows global network access to these drives. The user-specific entries of this file are stored in the directory.

See also

LOGDRIVE.INI (Page 292)

2.3.2 Connecting Drives with Display Machine Data

Configuring drives/computers

A maximum of four network drives can be set-up with the following display machine data:

MD 9676 CMM_DIRECTORY_SOFTKEY_PATH1="path1"

MD 9677 CMM_DIRECTORY_SOFTKEY_PATH2="path2"

MD 9678 CMM_DIRECTORY_SOFTKEY_PATH3="path3"

MD 9679 CMM_DIRECTORY_SOFTKEY_PATH4="path4"

The complete path must be entered. The paths are displayed on the softkeys.

The connections can be specified as DOS path e.g. "E:" or "G:\\VORBEREI\MPF_DAT" or as network addresses consisting of node identifiers, share name and directory, e.g. "\\R111\<Sharename>\MPF_DAT".

Example:

MD9677 CMM_DIRECTORY_SOFTKEY_PATH2=F:\R4711
 ;drive + directory
MD9678 CMM_DIRECTORY_SOFTKEY_PATH3=\\R4712\TEIL1\WELLE1.DIR
 ; computer name, enable name with directory data

MD9679 CMM_DIRECTORY_SOFTKEY_PATH4=\\R4713\Part2\MPF_DAT .DIR ; Computer name, enable name with directory data

The network connection to be specified in MD 9676 CMM_DIRECTORY_SOFTKEY_PATH1 is reserved for SINUMERIK ShopMill or ShopTurn.

Configuring texts for softkeys

Plain text identifiers for selecting network drive connections with softkeys can be stored as language-dependent user alarm texts under numbers 89901 to 89905 in a user text file that is known to the message block (see MBDDE.INI, [TextFiles] section). Text number 89901 correlates here to MD 9676 CMM_DIRECTORY_SOFTKEY_PATH1, etc.

Example:

MBDDE.INI:

...

[TextFiles]

USER_PLC=F:\DH\MB.DIR\MYTXT_

MYTXT_GR.COM:

89901 0 0 ""	Reserved
89902 0 0 "F:\R4711"	Softkey text for softkey 2
89903 0 0 "R4712/ MMC1"	Softkey text for softkey 3
89904 0 0 "R4713/ MMC2"	Softkey text for softkey 4
...	(2 blanks denote a line break)

Availability in the operating areas

The network drives are available in the following operating areas:

- In the MACHINE operating area on the expanded SK (softkey) bar behind the "Program overview" softkey
- In the PROGRAM operating area as an extended softkey bar in the main menu
- In the SERVICES operating area under "Manage data" (if 3 network drives are configured, 3 are displayed directly as horizontal softkeys, the fourth appears after pressing the ETC button)
- In the SERVICES operating area under "Data on" and "Data off" under the second vertical softkey "Network drives …"

The softkeys in the expanded softkey bar are assigned accordingly to MD 9676 (SK1) to 9679 (SK4) (i.e., with gaps if necessary). The configured text appears on the softkey. If no labeling text is configured for the softkey, the value of the corresponding machine data is displayed.

Assigning access authorization for drives/computers

Access authorizations are configured individually for each drive in the following display machine data:

For the **Program** and **Services:**

$MM_USER_CLASS_DIRECTORY1_P (9510)

$MM_USER_CLASS_DIRECTORY2_P (9511)

$MM_USER_CLASS_DIRECTORY3_P (9512)

$MM_USER_CLASS_DIRECTORY4_P (9513)

For the **Machine** area:

$MM_USER_CLASS_DIRECTORY1_M (9516)

$MM_USER_CLASS_DIRECTORY2_M (9517)

$MM_USER_CLASS_DIRECTORY3_M (9518)

$MM_USER_CLASS_DIRECTORY4_M (9519)

The display machine data for access authorizations are activated immediately, i.e., after switchover to the respective application, while a change in directories takes effect only after a restart.

Error messages

The configured network drives are evaluated when HMI-Advanced boots. This evaluation does not check whether the connection actually exists. This check is made after the drive is selected by the operator. If the configured path cannot be accessed, an error message is issued.

See also

MBDDE.INI (Page 298)

2.3.3 Connecting logical drives

Requirements

The "logical drives" softkey is available from access level 2 onwards (Service):

This is the reason that a machine operator cannot use this function, but instead only a service technician. A restart of HMI-Advanced is necessary to update the assignment of the softkeys.

With this softkey you can configure access to USB storage media on USB interfaces or PCU and TCU and to network drives (option) in the Programs, Services, or Diagnostics operating areas.

Function

The control can be connected to external network drives or other computers. This makes it possible, for example, to execute part programs from other computers.

Two alternatives are available:

- In the setting screen "Logic drives" under "Start-up" → "NC".
- By configuring file LOGDRIVE.INI.

Configuring a logical drive comprises the following tasks:

- Path name with the following structure:

  `"acttcu://<TCU connector>,<Partition>/<Path name>"`

  For the USB at a TCU name entry, connection field identifying the required connector, and a partition number
- and a softkey label (either a language-neutral or language-dependent softkey label by means of alarm texts).

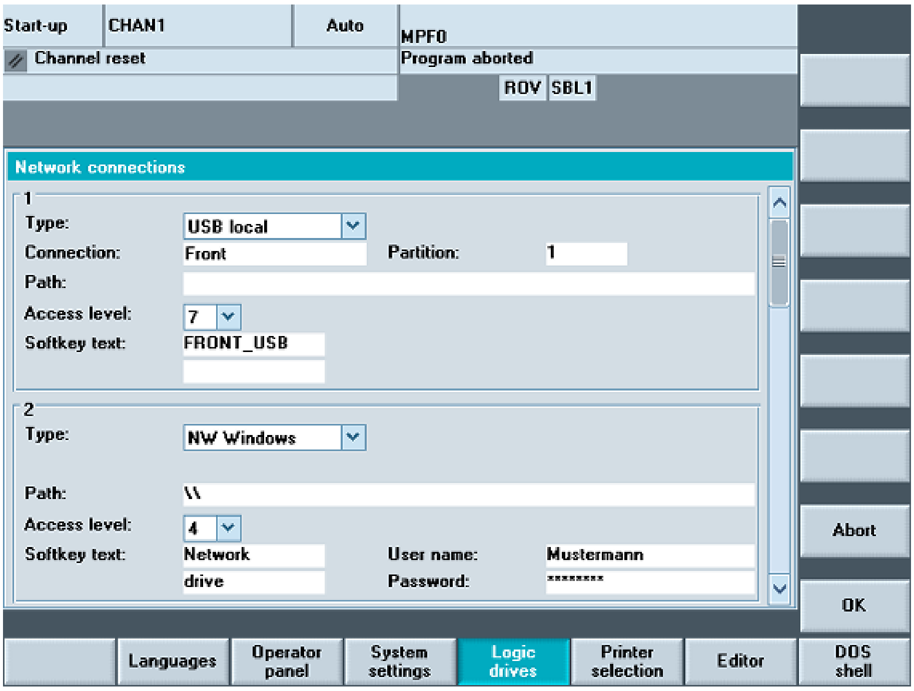
Softkey "USB local"

As supplied: Connection 1 is pre-assigned as "USB local".

The “"USB local" softkey is always displayed in the Machine, Program, and Services operating areas, even if it is not configured. This enables the front interface of the active TCU or the configured interface of the PCU to be displayed.

Note

If an OEM softkey (horizontal softkey 7) is configured, a maximum of 7 network drives are displayed in the program. The last entry is ignored.



Network connections

1
Type: USB local
Connection: Front
Partition: 1
Path:
Access level: 7
Softkey text: FRONT_USB

2
Type: NW Windows
Path: \\
Access level: 4
Softkey text: Network drive
User name: Mustermann
Password: ********

Figure 2-9 "Logical Drives" setting screen

Connection type / connection

The following connections can be selected.

No drive:	No additional fields are displayed:
Local USB:	Designation of the local USB interface: - Front (0.2 or 1:0.2 or FRONT) - X203 (1 or 1:1 or X203) - X204 (2 or 1:2 or X204)
Partition: (for local and global USB)	e.g. 1
Global USB:	Designation of the global USB interface: - Front (0.2 or 1:0.2 or FRONT) - X203 (1 or 1:1 or X203) - X204 (2 or 1:2 or X204)
Device: (only for global USB)	e.g. TCU1
NW Windows:	Designation of the network drive (under Windows). The path must begin with \\.
User name: (only for NW Windows)	The user name can be entered in the user interface, e.g. Mustermann.
Password: (only for NW Windows)	The password is displayed on the user interface with a *. It is encoded and entered in the logdrive.ini file.
Local drive:	Drive letter, e.g. D:

Access authorizations

The access authorizations to the connections can be assigned from level 0 to 7. If no entry is made, the default setting is applied (7 = Access for all).

Only one access level can be specified for all the operating areas:

- Machine
- Program
- Services

Softkey labels

Two lines with a maximum of 10 characters are available for the labeling text of a softkey. \n,%n or two blanks can be used to separate lines.

For labeling the softkeys, the alarm texts 89901- 89906 are evaluated in order to store language-dependent softkey texts if no other labeling is defined.

Data transfer to and from the TCU

The USB interface of the TCU that is currently active is always the one accessed, i.e. has the operator focus. If the active TCU changes during a data transfer, the data transfer remains active in the background.

If an operation (copy, delete, edit) is active on the drive at the time of the TCU switchover, the display is retained until the action is finished. The display is not updated until the operation is finished.

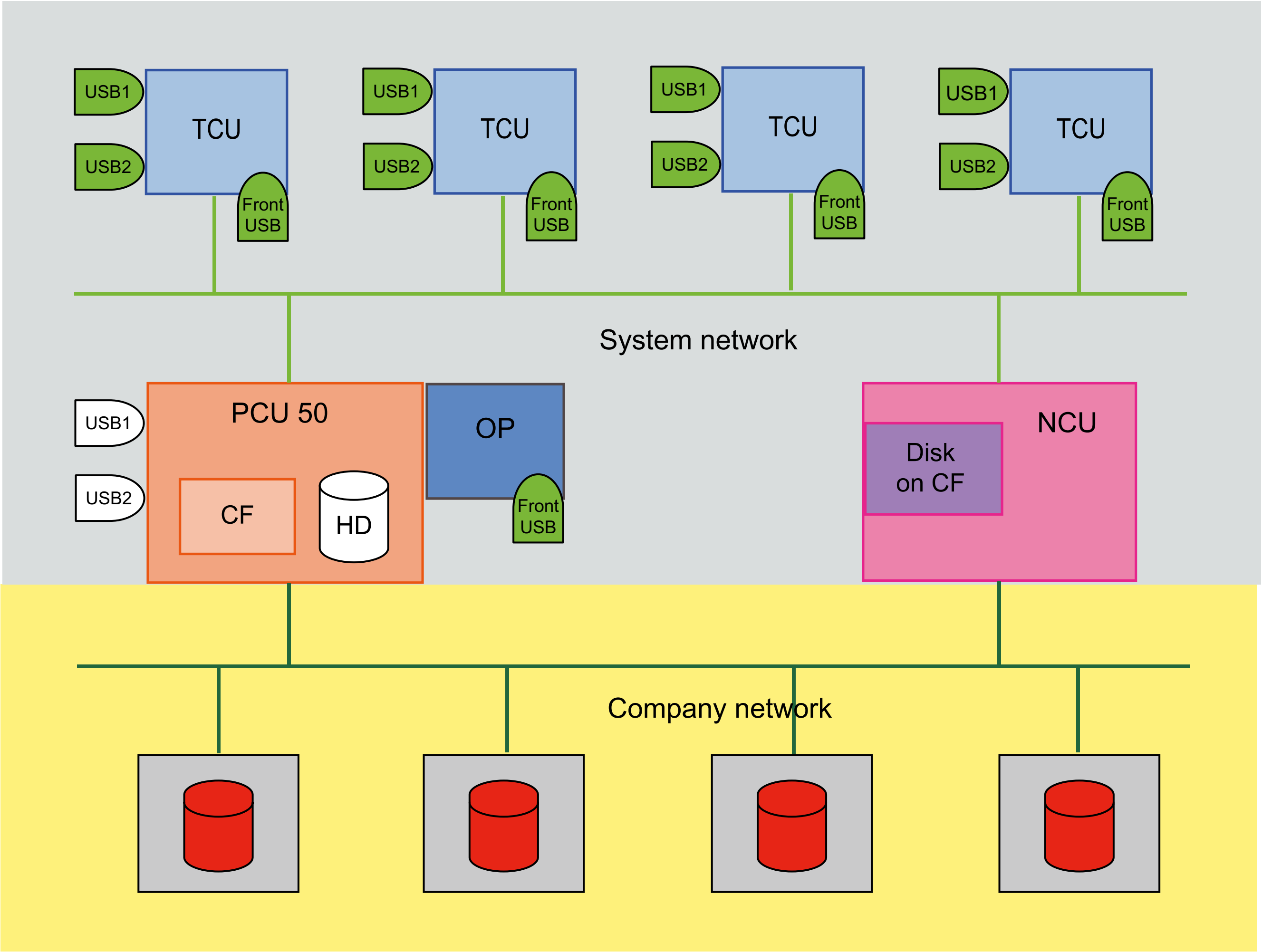
Note

Because the TCU is connected to the PCU via a network connection, the response times depend on the particular network. Executing a part program (EXTCALL) from USB memory is not recommended.

If a USB drive is disconnected while data are being written to or read from this drive, data loss can result.

"Network drives" option

The following overview clarifies when the network drives option is needed:



Legend:

🟩 Usable without option: all USB interfaces at the TCU

🟧 Usable without option: locally usable

⬜ Locally usable without option; "Remote" only usable with network drives option

🟥 Only usable with network drives option

🟪 Only usable with HMI user memory option, locally usable

Figure 2-10 Network drives option

2.4 Simulation

Overview

This function can be used to simulate machining on the screen.

Data evaluated by the simulation has to be copied into data management in the Dialog programming\Simulation data directory (\DP.DIR\SIM.DIR).

When the simulation program is started the following functions can be selected with the "Data matching" softkey:

- Match setup data
- Match tools
- Match machine data
- Match cycles

2.4.1 Display of the simulation files

Matching data

The files loaded into the simulation are displayed with softkey **Match data → Loaded data** All programs currently copied into the simulation are displayed. They are deleted from the simulation again when a different workpiece is selected.

Changes to these programs via the compensation editor are made directly in the HMI or NCK, and the modified programs are automatically copied into the simulation. If these programs are modified externally (e.g. via Services), the change is detected at the time of the next simulation selection and the programs concerned are copied into the simulation.

If files exist for the current workpiece (extension RPA, GUD, SEA, UFR, PRO, CEC, TOA, TMA), they are loaded into the simulation when the workpiece is selected. These files are retained when the workpiece is changed and their date stamp is monitored.

If these files are modified externally (e.g. via Services / Programs), the change is detected at the time of the next simulation selection and the files concerned are automatically loaded into the passive file system for the simulation.

- **Basic data**

  The initialization data (INITIAL.INI), tool data (TO_INI.INI) and definition files (*.DEF) loaded into the simulation are displayed. The files (CH(c)_xxx.INI, NC_xxx.INI and AX_SEA.INI) are displayed under "Loaded data".

  The date stamp for these files is evaluated. If these files are modified externally (e.g. via Services), the change is detected at the time of the next simulation selection and the files concerned are automatically loaded into the simulation.

- **Standard cycles**

  All standard cycles loaded into the simulation are displayed:

  The cycles are then copied into the simulation when required. Changes to the cycles are not detected automatically and must be detected by means of the "Match cycles" softkey.

- User cycles

  All user cycles loaded into the simulation are displayed:

  If the cycles have not been loaded before, the cycle interfaces must be defined in the file DPCUSCYC.COM. The cycles are then copied into the simulation when required. Changes to the cycles are not detected automatically and must be detected by means of the "Match cycles" softkey.
- Manufacturer cycles

  All manufacturer cycles loaded into the simulation are displayed:

  If the cycles have not been loaded before, the cycle interfaces must be defined in the file DPCUSCYC.COM. The cycles are then copied into the simulation when required. Changes to the cycles are not detected automatically and must be detected by means of the "Match cycles" softkey.

2.4.2 Data match of the simulation

Overview

The data to be evaluated by the simulation must be located in the data management system in \DP.DIR\SIM.DIR. They are downloaded each time simulation is started. Data are not activated until the simulation is restarted!

In addition, definition files in the data management directory are matched with the NC. If you do not wish the definition files in the data management directory to be matched with the NC, you can set this in DPSIM.INI.

```
[MAIN]
MATCH_NCU_ENVIRONMENT=0 ;only the INITIAL.INI is copied from the NC.
MATCH_NCU_ENVIRONMENT=1 ;(default)
```

Note

The data can be adjusted manually using the editor in the "Services" operating area, but if improper changes are made there is a risk that the simulation will not be able to start. You should therefore save the files systematically in case the data should not/cannot be retrieved from the NC.

The following functions are listed under the softkey "Data matching":

Match setup data

The following data are listed from which a selection can be made as to which data is downloaded from the active file system of the NC into the simulation:

- R parameters (RPA)
- User data (GUD)
- Global user data
- Work offsets (UFR)

- Global work offsets
- Setting data (SEA)
- Global setting data
- Axis-specific setting data
- Protection zones (PRO)
- Sag/angularity (CEC)

The data can be selected in any combination. The channels are selected using softkeys (if there is more than one channel for simulation); the current channel of the simulation is offered as default. Channel-specific data are only offered if the selected channel is present on the NCK. The selected data are copied into corresponding files in the data management system under \DP.DIR\SIM.DIR using the "OK" softkey and loaded into the simulation.

Existing files are overwritten without prompting.

If data are selected for all channels, any associated channel-specific files that are present are deleted; channel-specific files are always created or overwritten.

Before the data are loaded into the simulation, a "channel reset" is performed for the simulation.

If an error occurs when copying from the NCK, the transfer can be repeated or aborted (dialog box: Abort/Repeat). If an error occurs when loading into the simulation or a simulation alarm is triggered, the transfer is aborted.
Pressing the "OK" softkey saves the current data selection and displays it again the next time the dialog box is opened.

Match tools

The NC tool data for all channels are copied into the TO_INI.INI file and then activated in the simulation. Existing tools are removed beforehand. The function is offered automatically whenever simulation is selected if tool data in the NC have changed.

Match machine data

All definition files (UMAC.DEF, SMAC.DEF, MMAC.DEF, UGUD.DEF, SGUD.DEF, MGUD.DEF, GUD4.DEF – GUD9.DEF) that are active in the NC are copied to the corresponding files in the HMI. Files that are not active in the NCK are deleted from the HMI. All files (CH(c)_xxx.INI, AX_SEA.INI, NC_xxx.INI) are deleted with the exception of the tool data.

The machine data from the INITIAL.INI file are interpreted by the simulation. We recommend to set the following machine data:

```
MD 11210: UPLOAD_MD_CHANGES_ONLY = 0
```

Match cycles

Cycle changes are not monitored by the simulation.

They are either copied automatically into the simulation as required (PRELOAD/CYCLES=0) or copied in their entirety into the simulation when the simulation boots.

If a cycle is changed, it must be copied into the simulation using the "Match cycles" softkey. Only cycles that have changed since they were last accessed are copied into the simulation (along with newly imported cycles).

Cycles in the simulation that have since been deleted are deleted from the simulation. If an automatic cycle interface is not used, the simulation must be restarted.

Mirroring tools

Differently oriented tool carriers (e.g. in different NC channels) can be used in the same machine tool, for which the TOA length compensation parameters are subject to machine tool-specific frame transformations (e.g. MIRROR) and have to be specified differently from the usual settings.

In order to be able to support this constellation when orientating the graphic tool, the SIM graphic module DPSMASCH.DLL among others, is expanded, in the turning version, by an essentially universal evaluation of the TOA parameter.

e.g. $TC_DP3[n,m]=... can also be evaluated with negative values with the convention so that as standard **positive length dimensions** generally result in a machining "behind the center of rotation" and **negative length dimension** results in machining "in front of the center of rotation".

Special NC instructions may also be needed in the corresponding user cycles for tool carrier selection to ensure that the simulation proceeds with the correct orientation of the graphics tools, e.g.

```
...
IF $P_SIM
 $SC_MIRROR_TOOL_LENGTH=1 ; tool length compensation
mirror mirrored axes for SIM
ENDIF
...
```

Operator control (milling and turning):

The sign-compliant evaluation of the TOA parameters at the interface to the graphics module means that no operator action is required.

Rotating the coordinate system

The orientation of the coordinate system for simulation views (until now largely incomplete in the turning version) can be altered and adjusted, for example, to rotary machines.

For this purpose, the INI entry [VIS] PLANE_VIEW=.. (in ..\user\dpmwp.ini for the milling version or dptwp.ini for the turning version) can be modified.

In order for advantage to be taken of these possibilities in machines that differ from the standard orientations, the value range of the entry is published (B/F: back/front, U/D: up/down, L/R: left/right) and the functionality during turning expanded accordingly (incl. the derived views: Outside cut, half cut and full cut, surface area) (wire model is kept as before!).

In some cases (lathes) the entries can be directly accessed at the user interface:

```
a[VIS]
…
;Plane viewpoint default for standard milling: BDL(BackDownLeft)
;possible BDL, BUL, BUR, BDR, FDL, FUL, FUR or FDR PLANE_VIEW=BDL
```

or

```
[VIS]
…
;Plane viewpoint default for standard horizontal lathes:
BDR(BackDownRight)
;for vertical lathes with +Z upward: BDL or with +Z downward: FUL
(*** from SW5.3 ***)
;possible BDL, BUL, BUR, BDR, FDL, FUL, FUR or FDR
PLANE_VIEW=BDR
```

Note

Entries other than BDR, BDL or FUL are of no use for lathes at present.

Operator control (only turning):

To support vertical lathe orientations, the existing "Set spindle" form has been expanded to include an additional "Machine orientation" frame group, initially with 3 option controls:

"horizontal standard" (default: PLANE_VIEW=BDR),

"vertical, downward feed" (PLANE_VIEW=BDL)and

"vertical, upward feed" (PLANE_VIEW=FUL).

Using the "Change default" softkey the adjustments made here can also be used as a permanent reference for all new workpieces.

Optimizations

- **Input monitoring "Unwind diameter peripheral surface"**

  In the turning version, in conjunction with the blank inputs, the consistency of the "External diameter" and "Unwinding diameter surface area" parameters is checked and, if necessary, adjusted automatically (e.g. if the external diameter is reduced), and missing entries are signaled (e.g. empty input field for unwind diameter).

- **Tool center point path ("without tool data")**

  Temporarily received for the subsequent selection of a tool data source (e.g. "with tools").

  In order for the correct selection and deselection of tool compensations (LK, FRK, SRK) to be checked visually, it is useful to be able to visually compare the tool center-point path when tool compensations are deselected and the tool path when tool compensations are selected.

  For this purpose, the tool center-point path produced from a simulation run "without tool data" is therefore generally retained for the first subsequent simulation run "with tool data" (tool paths with and without tool compensations are overlaid graphically for one SIM session). Thereafter the tool paths are reset as usual at the start of each simulation.

2.4.3 Speeding up the simulation boot

Simulation start

The simulation application is normally started only on request by the operator. This means that the system is only loaded as required as a result of the software components to be downloaded.

At the same time, however, this significantly increases the start time for the simulation when it is called for the first time. The start time is the time from selecting the simulation to execution of the first NC command ("visible" result).

Downloading when the HMI boots

To speed-up simulation starting, the downloading times can be transferred to the HMI boot. The settings for this are carried out in the REGIE.INI file (OEM or USER directory).

```
[StartupConfiguration]
Startup12 = name:=dpncsim, Timeout:=0
[TaskConfiguration]
Task27 = name := dpsim, Timeout := 30000
```

Note

The entry used in the [StartupConfiguration] section must not have been used anywhere else. Check this in the REGIE.INI file in the MMC2, ADD_ON, OEM and USER directories.

Loading when SimNCK boots

Configuration files for the NC environment (tool data, cycles) are loaded when the SimNCK is booted. This means that only the program to be executed and its subroutines need to be loaded into the simulation NCK. This behavior when booting is controlled by means of the [PRELOAD] and [MAIN] sections in the DPSIM.INI file.

The starting time of the simulation can be shortened by using the following settings:

```
[PRELOAD]
CYCLES=1
```

All cycles (from NCK and hard disk) are preloaded when the simulation NCK is booted with DPNCSIM.EXE.

```
TOOLS=1
```

The files for the tool mapper (TO_INI.INI and TO_ADDON.INI) are preloaded when the simulation NCK is booted (if applicable).

2.4.4 Expanding geometrical tool data

Representation of the tools

Previously tools could only be shown with their cutting edge, shaft and holder, which could be calculated from the data in the NCK.

With the expansion of the tool geometry data, it is now possible to show only the cutting edge ("flying edge") in the simulation image, without the shank and holder. Additional, deviating parameters can also be incorporated into the simulation on a tool-by-tool basis.

The tool data are defined in the file **SIMTOGEO.INI**, which is stored in one of the directories \USER, \OEM, \ADD_ON, or \HMI_ADV. This file is not supplied with the system as standard.

Other requirements:

- Activating the "flying edge" function in DPTWP.INI or DPMWP.ini
- Machine data: Defining memory for OEM parameters
- Creating the SIMTOGEO.INI file, insert link in file

Parameterization

In the SimToGeo.ini file in the section [Settings] there must be an entry that establishes a link to the NCK tool data.

```
[Settings]
Link=TC_DPCx with x = 1... 10 or
Link=TC_DPCSx with x = 1... 10
```

We recommend using the parameter TC_DPCS1 as the link entry for new projects, as this has been specially reserved for this purpose. A different link setting should only be used if the parameter TC_DPCS1 is already in use in an existing project.

Example

With Link=TC_DPCS1, for example, for every tool for which entries exist in SimToGeo.ini, an entry $TC_DPCS1[t, d]= <Identifier> must be specified for the NCK tool data, where t = tool number and d = duplo number. Identifier must be noted as a real number, e.g. 500.1, where the number before the decimal point can represent a tool type and the number after the decimal point allows a further distinction.

The associated values in SimToGeo.ini appear below the corresponding identifier:

```
[500.1] ; roughing tool 1
tool_type = 500
...
```

Optional parameters for orienting milling cutters

Tools are still displayed in the 3 main planes, XY, YZ or ZX. If a tool cannot be displayed, a polymarker is displayed.

Only **one** of these entries may be used:

```
TC_DPV3= ; 1 = +X, -1 = -X
TC_DPV4= ; 1 = +Y, -1 = -Y
TC_DPV5= ; 1 = +Z, -1 = -Z
```

It is not possible to enter a tool orientation that is freely located in space.

In the file SimToGeo.ini, a tool can be defined under tool_type that differs from the definitions of the NC. If the corresponding cutting edge parameters (e.g. $TC_DPCS1[t,d]) can be set, a link to or evaluation of geometrical data with SimToGeo.ini is also possible.

A complete sample file with entries for each permissible tool type can be found in Chapter "SIMTOGEO.INI"

Setting machine data

MD 18080: MM_TOOL_MANAGEMENT_MASK='H4'	;Set aside memory for OEM data
MD 18204: MM_TYPE_CCS_TOA_PARAM[0]=4	;Data type 4 = Real
MD 18206: MM_NUM_CCS_TOA_PARAM=1	;Number of OEM data per cutting edge 0 to 10

Procedure when commissioning the system:

1. Activating the "Flying edge" function in DPTWP.INI or DPMWP.INI, depending on the turning/milling technology:

   ```
   [SETUP]
   ;*** only SW6.4 or later ***
   ;Display tool only as cutting edge "flying edge" (ON/OFF)
   TOOL_AS_CUTTING_EDGE=ON ;default = OFF
   ```

2. Defining machine data or memory for OEM parameters:

   `$MN_MM_TOOL_MANAGEMENT_MASK=4 ;for SimToGeo.ini, set bit 2=1`

   If MD is already pre-assigned, the bit must be additionally set - define memory and type for SIEMENS-OEM parameter $TC_DPCS1 `$MN_MM_NUM_CCS_TOA_PARAM=1 ;TC_DPCS1 is reserved.`

   `$MN_MM_TYPE_CCS_TOA_PARAM[0]=4 ;data type float for TC_DPCS1`

   or

   Define memory and type for USER-OEM parameter $TC_DPC1

   ```
   $MN_MM_NUM_CC_TOA_PARAM=1 ;TC_DPC1 is reserved
   $MN_MM_TYPE_CC_TOA_PARAM[0]=4 ;data type float for TC_DPC1
   ```

3. Set up SimToGeo.ini:

   Define link entry for Siemens or User OEM parameter and expand geometrical data. This is done using the OEM variable ($TC_DPCS1/$TC_DPC1). The values of these variables determine the geometrical block that is used.

   For example, the entry for tool 1 cutting edge 1 (T1 D1) $TC_DPCS1[1,1]=500.1 would be referred to the geometrical data block [500.1] in the SimToGeo.ini file. It is possible for more than one tool to refer to the same geometrical data block.

4. Adding expanded geometrical tool data. This in done in blocks, which are distinguished e.g. using identifiers, float type.

   For example [500.1].

   ```
   [Settings]
   Link=TC_DPCS1 ;link to TO_INI.INI, Siemens OEM
   ;Link=TC_DPC1 ;link to TO_INI.INI, USER OEM

   [500.1] ;T1 D1 roughing tool type 500 SL3 L1=100, L2=55, R1
   tool_type=500 ;tool type then $TC_DP1 is not required
   insert_length=10 ;TC_DP8, plate length
   holder_angle=105 ;TC_DP10, holder angle, not rake angle
   reference_direction=3 ;TC_DP11, cutting edge position
   clearance_angle=5 ;TC_DP24, clearance angle
   ```

5. Link to tool data/TO_INI.INI:

   ```
   CHANDATA(1)
   $TC_DPCS1[1,1]=500.1 ;Siemens OEM T1 D1
    Roughing tool SL3 type 500
   $TC_DPC1[1,1]=500.1 ; User OEM
   $TC_DP1[1,1]=500
   $TC_DP2[1,1]=3
   $TC_DP3[1,1]=100
   $TC_DP4[1,1]=55
   ```

2.4.5 Optimizing the memory requirement

Function

To avoid having to carry out memory settings in NC machine data if additional memory is needed for the simulation, these settings can be carried out in the DPSIMNCK.INI file in the [DP_SIMNCK] section.

The default values for the simulation are entered in the dpsimnck.ini file in the [DP_SIMNCK] section, generally eliminating the need for additional memory settings in the NC. **These offset values have a cumulative effect on machine data with the same name.**

Entries in the file DPSIMNCK.INI

Negative values are ignored.

The following entries with default values are contained in the [DP_SIMNCK] section:

```
...
[DP_SIMNCK]

DP_SIMNCK_MM_NUM_GUD_MODULES_OFFSET = 0
```

with MD 18118: MM_NUM_GUD_MODULES

```
DP_SIMNCK_MM_NUM_GUD_NAMES_NCK_OFFSET = 5
```

with MD 18120: MM_NUM_GUD_NAMES_NCK

```
DP_SIMNCK_MM_NUM_GUD_NAMES_CHAN_OFFSET = 30
```

with MD 18130: MM_NUM_GUD_NAMES_CHAN

```
DP_SIMNCK_MM_GUD_VALUES_MEM_OFFSET = 20
```

with MD 18150 MM_GUD_VALUES_MEM

```
DP_SIMNCK_MM_NUM_USER_MACROS_OFFSET = 10
```

MD 18160: MM_NUM_USER_MACROS

```
DP_SIMNCK_MM_NUM_MAX_FUNC_NAMES_OFFSET = 40
```

with MD 18170 MM_NUM_MAX_FUNC_NAMES

```
DP_SIMNCK_MM_NUM_MAX_FUNC_PARAM_OFFSET = 350
```

with MD 18180 MM_NUM_MAX_FUNC_PARAM

```
DP_SIMNCK_MM_MAX_SIZE_OF_LUD_VALUE_OFFSET = 400
```

with MD 18242: MM_MAX_SIZE_OF_LUD_VALUE

```
...
```

2.5 Configuring user alarms

2.5.1 Classifying user alarms

Number ranges

The following alarm numbers are available for the cycle alarms, compile cycle alarms, and PLC alarms:

Numerical range	Designation	Effect	Deletion
60000 - 60999	Cycle alarms (Siemens)	Display, interlocking NC start	Reset
61000 - 61999		Display, interlocking NC start, no motion after executing the pre-decoded blocks	Reset
62000 - 62999		Display	Cancel
63000 - 64999	Reserved		
65000 - 65999	Cycle alarms (user)	Display, interlocking NC start	Reset
66000 - 66999		Display, interlocking NC start, no motion after executing the pre-decoded blocks	Reset
67000 - 67999		Display	Cancel
68000 - 69000	Reserved		
70000 - 79999	Compile-cycle alarms:		
400000 - 499999	PLC alarms general		
500000 - 599999	PLC alarms for channels		
600000 - 699999	PLC alarms for axes and spindles		
700000 - 799999	PLC alarms for users		
800000 - 899999	PLC alarms for sequence cascades/graphs		

Reference: Function Manual Basic Functions: PLC basic program (P3 sl)

Selecting the language for alarm texts

If alarm texts are to be configured in languages other than those already available in the control, then a softkey "language selection" can be configured. The name of the text file is used to assign language-specific user alarm texts. The appropriate code and the extension .com are added to the user file name entered in MBDDE.INI.

Supported languages

Table 2- 2 Languages and code page/ANSI table used

Language	⇓ Abbreviation	Code page (DOS)	ANSI table (Windows)
Czech	CZ	852	1250
Danish	DK	850	1252
Finnish	FI	850	1252
French	FR	850	1252
German	GR	850	1252
Hungarian	HU	852	1250
Italian	IT	850	1252
Dutch	NL	850	1252
Polish	PL	852	1250
Portuguese	PO	850	1252
Russian	RU	855	1251
Spanish	SP	850	1252
Swedish	SW	850	1252
Turkish	TR	857	1254
English	UK	850	1252
Japanese	JA	932	-
Chinese (Simplified)	CH	936	-
Korean	KO	949	-
Chinese (Traditional)	TW	950	-

Installing languages

Languages that are not included with the standard product must first be installed. An additional tool is implicitly required for Asian languages (e. g. NJWIN). The appropriate fonts are installed automatically with this.

When each language module is installed, the tool also installs the appropriate fonts, and all necessary entries are made in the MMC.INI file and in the registry.

When commissioning the system, the DLLs, text files (machine data files) and language-dependent INI files are stored in the "\mmc2\language" directory, and the alarm texts are stored in the “\dh\mb.dir” directory.

Assignment between ANSI tables and fonts

Table 2- 3 ANSI table 1250 (Central Europe)

Font	File name	Name of font
Arial	Cearial.ttf	Arial CE (True Type)
Arial bold	Ceariabd.ttf	Arial CE Bold (True Type)
Arial italic	Ceariali.ttf	Arial CE Italic (True Type)
Arial bold italic	Caeriabi.ttf	Arial CE Bold Italic (True Type)

Table 2- 4 Ansi table 1251 (Cyrillic)

Font	File name	Name of font
Arial	Aricyr.ttf	Arial Cyr (True Type)
Arial bold	Aricyb.ttf	Arial Cyr Bold (True Type)
Arial italic	Aricyri.ttf	Arial Cyr Italic (True Type)
Arial bold italic	Aricyrbi.ttf	Arial Cyr Bold Italic (True Type)

Table 2- 5 ANSI table 1252 (West Europe)

Font	File name	Name of font
Arial	Windows default	Arial (True Type)
Arial bold	Windows default	Arial Bold (True Type)
Arial italic	Windows default	Arial Italic (True Type)
Arial bold italic	Windows default	Arial Bold Italic (True Type)

Table 2- 6 ANSI table 1254 (Turkish))

Font	File name	Name of font
Arial	Trar.ttf	Turkish Arial (True Type)
Arial bold	Trarbd.ttf	Turkish Arial Bold (True Type)
Arial italic	Trari.ttf	Turkish Arial Italic (True Type)
Arial bold italic	Trarbi.ttf	Turkish Arial Bold Italic (True Type)

Online help for alarms

Creating user-specific online help texts is described in the /HE1/ Commissioning Manual, online help.

See also

Configuring the '"Language Selection" softkey (Page 28)

2.5.2 Configuring user alarms

Files supplied with the system

The files with the alarm texts in the ASCII format are saved on the PCU hard disk under the following path:

MMC	F:\dh\mb.dir\alm_XX.com
NCK	F:\dh\mb.dir\aln_XX.com
PLC	F:\dh\mb.dir\alp_XX.com
ZYK	F:\dh\mb.dir\alz_XX.com
CZYK	F:\dh\mb.dir\alc_XX.com

In these file names, "XX" stands for the code of the appropriate language. The **standard files** should **not be changed** by the user in order to store his or her own alarm texts.

Note

To prevent a modified MBDDE.INI file from being overwritten when the software is updated, it must be stored in the path designated for that purpose: ..\user\mbdde.ini

Creating user alarms

The user can replace the alarm texts supplied by his own texts or add his own texts. To do this, use the Editor available in the area "Start-up" → "HMI" → "Editor".

To do this, the additional files must be created in directory **f:\dh\mb.dir** (MBDDE alarm texts) via the "Services" operating area.

The texts from the user files overwrite standard texts with the same alarm number. Alarm numbers that do not already exist in the standard texts are added.

Format of text file for cycle alarm texts

The structure of the text file for cycle alarms and compile cycle alarms is as follows:

Alarm number	Display	Help ID	Text or alarm number
60100	1	0	"No D number %1 is programmed"
60101	1	0	60100
...	...	...	...
65202	0	1	"Axis %2 in channel %1 is still moving"
// Alarm text file for cycles in English			

The number range in the list is not available with every number.

Editor

Any ASCII editor (e.g. DOS editor) can be used for editing files. When editing the text files with a different editor, make sure that they are then saved in ASCII format.

Note

Changes to the alarm texts only take after the HMI has restarted. When creating text files, make sure that the correct date and time are set on the HMI system. Otherwise, the user texts may not appear on the screen.

Activating the alarm text files

The alarm texts generated by the user are activated in the file MBDDE.INI in the Section [Textfiles] by the following entries:

```
...
[Textfiles]
MMC=F:\dh\mb.dir\alm_
NCK=F:\dh\mb.dir\aln_
PLC=F:\dh\mb.dir\plc_
ZYK=F:\dh\mb.dir\alz_
CZYK=F:\dh\mb.dir\alc_
UserMMC=
UserNCK=
UserPLC=
UserZyk=
UserCZyk=
...
```

Example for two additional files (texts for PLC alarms, modified NCK alarm texts) in the MBDDE.INI file:

```
...
User MMC =
User NCK = F:\dh\mb.dir\mynck_
User PLC = F:\dh\mb.dir\myplc_
User ZYK =
User CZYK =
...
```

Language-specific nature of alarm texts

The name of the text file is used to assign languages to user alarm texts. The appropriate code and the extension .com are added to the user file name entered in MBDDE.INI. The language codes are to be found in section "Supported languages" (Table 2-2).

If alarm and message texts are created externally, they must be generated with the listed code pages, ANSI tables and fonts (tables 2-3 to 2-6).

The alarm and message texts are stored in the MBDDE alarm texts directory.

1. Convert the old text file to an ANSI text file using "Save As…" in the Windows Editor (WinWord, Notepad, etc.).
2. Load the new text file and converted old text file into the Windows editor and merge them.
3. Save the newly generated text file and insert it into the directory "MBDDE alarm texts".

See also

MBDDE.INI (Page 298)

2.5.3 Example: Extending an alarm text for user PLC alarms

Overview

Freely definable user texts can be added to user PLC alarms to facilitate evaluation of diagnostics information when an error has occurred.

Procedure

The following entry is in the configuration file for the alarm server (file MBDDE.INI) in the section [IndexTextFiles]:

Extract from MBDDE.INI:

```
[IndexTextFiles]
ALNX=f:\dh\mb.dir\alnx_
ALSI=f:\dh\mb.dir\alsi_
DUMMY=f:\dh\mb.dir\dummy_
```

1. First of all, the file containing the alarm text extension must be created, or one of the available ALNX (for extending NCK alarm texts) or ALSI (for extending safety alarm texts) must be copied and then entered in the section [IndexTextFiles] along with the path.

   This is file dummy_gr.com in our example.

   ```
   000000 0 0 "Valve 1"
   000001 0 0 "Valve 2"
   000002 0 0 "Valve 3"
   000003 0 0 " ... "
   000004 0 0 " ... "
   ```

   The structure is exactly the same as for the alarm text files (with which you are already familiar), apart from the fact that the parameter value transferred from the PLC is now what used to be the alarm number.

Auxiliary value	Display	Help ID	Alarm text
000000	0	0	"Valve 1"
000001	0	0	"Valve 2"
000002	0	0	"Valve 3"
000003	0	0	"Valve 4"

1. The alarm text extension is inserted into the alarm text file (e.g. alp_gr.com) as follows:

   ```
   700000 0 0 "Attention: %1<DUMMY> failed!"
   ```

   The %1 parameter evaluates the first parameter transferred from the PLC and is linked to the "dummy_gr.com" file with <DUMMY>. This was done during the previous steps (the entry in MBDDE.INI).

2. Calling the auxiliary value from the PLC via SFC17/18:

   ```
   CALL SFC 18
   SIG :=
   ID :=
   EV_ID :=
   SD :=
   RET_VAL:=
   ```

Parameter	Declaration	Data type	Memory area	Description
SIG	INPUT	BOOL	I, Q, M, D, L	The signal triggering the message
ID	INPUT	WORD	I, Q, M, D, L	Data channel for messages: W#16#EEEE
EV_ID	INPUT	DWORD	I, Q, M, D, L	Message number (not permitted: 0)
SD	INPUT	ANY	I, Q, M, D, T, C	Auxiliary value • Maximum length: 12 bytes • Only BOOL data types are permitted (not permitted: bit array), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Error information

(Please take the exact structure of the SFC17/18 from the STEP 7 online help!)

Result

If the value "1" is transferred, the following message will be displayed:

```
700000   Attention: Valve 2 has failed!
```

Parameterizing machine data 3

3.1 Handling machine data via the HMI

Overview

Specific settings can be made to address manufacturer manufacturer and user requirements by:

- Configuring the HMI operator interface using INI files
- Parameterizing the control using machine data
- Programming PLC Functions

> ⚠ DANGER
>
> 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.

Requirements

Access to the machine data operating area can be controlled by key-operated switch or password.

Classification of machine data

The machine data is divided into the following areas:

- General machine data ($MN)
- Channel-specific machine data ($MC)
- Axis-specific machine data ($MA)
- Display machine data ($MM)
- Drive-specific machine data ($M_)
- Machine data control unit: Drive parameters (solution line)
- Infeed-machine data: Drive parameters (solution line)

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

The following properties of the machine data are displayed from left to right:

- Machine data number
- Machine data name, with field index if applicable.
- Value of the machine data

- Unit of the machine data
- Effectiveness

Effectiveness of the machine data

The right-hand column indicates when a machine data becomes effective:

so = immediately effective
cf = when confirmed via the "Activate MD" softkey
re = reset
po = POWER ON (NCK Power On reset)

Physical units of the machine data

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

Display	Unit	Measured quantity
m/s**2	m/s²	Acceleration
rev/s**3	rev/s³	Acceleration change for the rotating axis
kg/m**2	kgm²	Moment of inertia
mH	mH (millihenry):	Inductance
Nm	Nm (Newton meters):	Torque
us	µs (microseconds):	Time
µA	µA (microamperes):	Current magnitude
µVs	µVs (microvolt-seconds):	Magnetic flux
userdef	user-defined:	The units are defined by the user.

If the machine data does not use units, no units are displayed.

If the data are not available, the "#" symbol is displayed instead of the value. If the value ends in an "H", it is a hexadecimal value.

3.1.1 Setting-up a display filter for machine data

Function

The purpose of the display filters offered for “General machined data” is to selectively reduce the number of machine data displayed.

For this purpose, all machine data are assigned to certain display groups, e.g. configuration data. The following applies:

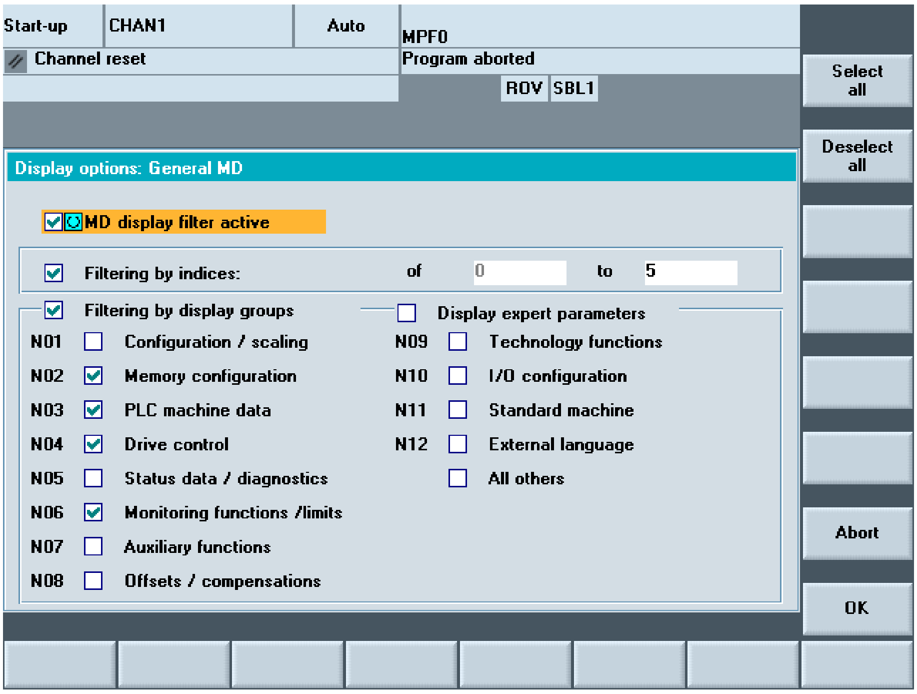
- Each area has its own group organization.
- Each group corresponds to one bit in the word filter (previously a reserve)

- Each area has a maximum of 13 groups (group 14 is reserved for Expert parameters, bit 15 is reserved for add-ons).
- Display machine data is subdivided into groups.

Display filter active

If the display filter is active, only the set display groups are displayed. With "All others" only the machine data are displayed for which no filter bit is set. The softkeys behave in the same manner as with the display options for general machine data. The settings are saved.

Example


Start-up CHAN1 Auto MPF0

Channel reset

Program aborted

ROV SBL1

Display options: General MD

☑ MD display filter active

☑ Filtering by indices: of 0 to 5

☑ Filtering by display groups

□ Display expert parameters

N01 □ Configuration / scaling

N02 ☑ Memory configuration

N03 ☑ PLC machine data

N04 ☑ Drive control

N05 □ Status data / diagnostics

N06 ☑ Monitoring functions /limits

N07 □ Auxiliary functions

N08 □ Offsets / compensations

N09 □ Technology functions

N10 □ I/O configuration

N11 □ Standard machine

N12 □ External language

□ All others

Figure 3-1 Display filter for machine data

Initialization

When you open a machine data window, the filter setting that matches the area is automatically updated.

This filter allows you to hide machine data and only show selective machine data as "display options" according to their specific field of use. For example, machine data can be grouped according to their affiliation.

Filter criteria

The following table shows the criteria for displaying machine data in the order in which they are evaluated:

Criterion	Checks
1. Access authorizations	If the level of access authorization is not sufficient, the MD is not displayed. Otherwise criterion 2 is checked.
2. Display filter active	The MD is always displayed when the filter is not active. Otherwise criterion 3 is checked.
3. Expert parameters	The MD is not displayed if expert parameter bit is set and expert parameters is not selected. Otherwise criterion 4 is checked.
4. Display groups	If at least one group bit is both set and selected in the display filter, criterion 6 is checked. Otherwise criterion 5 is checked.
5. All others	If none of the group bits is set and "All others" is selected in the display filter, then criterion 6 is checked. If none of the group bits is set and "All others" is not selected in the display filter, then the MD is not displayed.
6. Indices from ... to ...	If the index check is selected and the index of an array is within the chosen range, then the MD is displayed. If the index check is selected and the index of an array is not within the chosen range, then the MD is not displayed. If the index check is not selected, then the MD is displayed.

3.1.2 Creating user views

User views

User views are user-specific groups of machine data. They are used to call all relevant machine data in a certain operating state from various areas for processing.

Overview of machine data

You can include the following machine data in the user view:

- General machine data
- Channelspecific machine data
- Axis-specific machine data
- General setting data
- Channel-specific setting data
- Axis-specific setting data

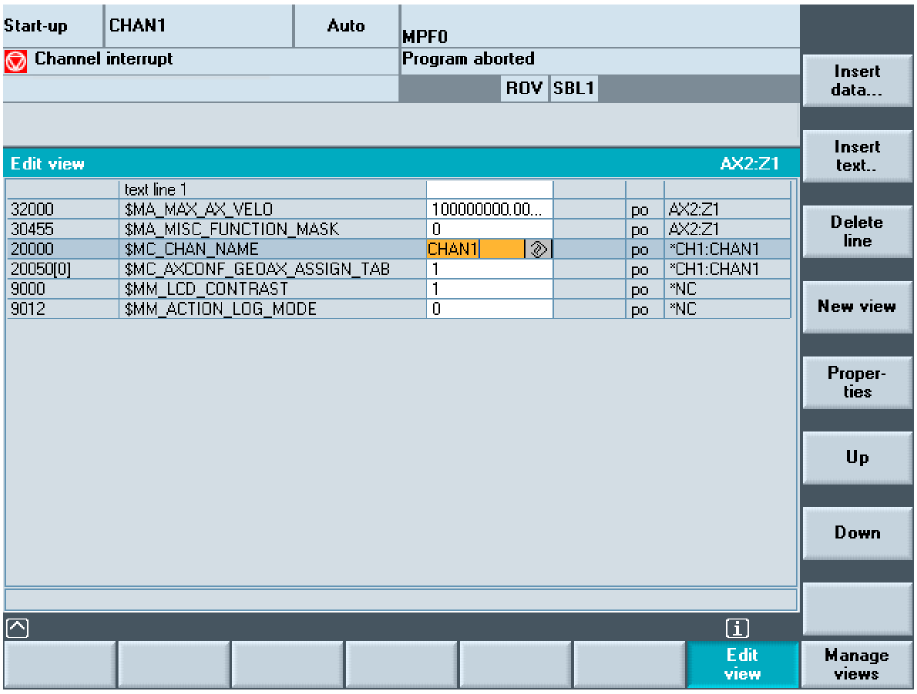
- General display machine data
- Channel-specific display machine data
- Machine data for SIMODRIVE drives (powerline)
- Machine data for SINAMICS drives, for Control Units, I/O modules and infeed

Example

Creating user view TEST01:

1. Select "New View". An empty window is opened in order to insert the appropriate machine data and comment lines.
2. In order to insert a comment, select "Insert Text...", and enter a text, for example, "text line 1". Acknowledge using the <INPUT> key in order to accept the text.
3. In order to insert a machine data, select "Insert Data...". Select the machine data area, e.g. "Axis-specific machine data". The corresponding range of numbers 30 000 - 38 000 is displayed in the list:
4. Mark machine data 32 000. In order to transfer the machine data into the user view, press the softkey "Insert before/after line".
5. For additional machine data, proceed in the same way or user "Search": Within a range of numbers, you can also mark and insert a machine data directly using "Search".
6. In order to complete the selection, press the softkey "<<" to return to the window "Edit View".

Result:

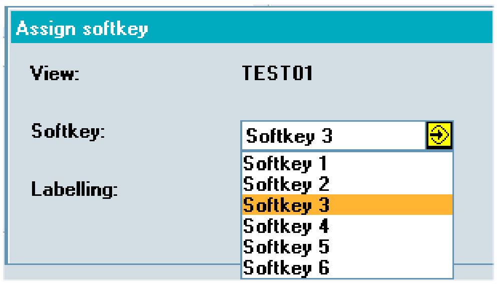


	text line 1				
32000	$MA_MAX_AX_VELO	100000000.00...		po	AX2:Z1
30455	$MA_MISC_FUNCTION_MASK	0		po	AX2:Z1
20000	$MC_CHAN_NAME	CHAN1		po	*CH1:CHAN1
20050[0]	$MC_AXCONF_GEOAX_ASSIGN_TAB	1		po	*CH1:CHAN1
9000	$MM_LCD_CONTRAST	1		po	*NC
9012	$MM_ACTION_LOG_MODE	0		po	*NC

Figure 3-2 Example: User view

The following softkeys are available to edit the user view:

1. You can shift lines using the softkeys "Up" or "Down" - correspondingly, lines are deleted with "Delete line".
2. Select "Properties" in order to enter descriptive text for the machine data or to modify existing text. The descriptive text is displayed at the bottom in the info line, e.g. "Channel name".
3. To save the user view, select "Manage views". Enter a name, e.g. Test01 and accept it using the <INPUT> key.
4. In order to display the user view, select "Assign Softkey" and enter a text for the softkey labeling.



Assign softkey

View: TEST01

Softkey: Softkey 3

Labelling:

Figure 3-3 Assigning softkeys for direct selection

Note

There are 6 softkeys that can be used to saved user views.

In the "Axis" field it is possible to enter the axis number or select the appropriate axis from a list.

Axis number or axis name is marked with a "*" in the user view until their assignment changes.

3.2 Parameterizing the control using machine data

3.2.1 Machine data for setting actual value, scratching, PRESET

Access levels for system frames

The machine manufacturer or user can assign the access levels for system frames as well as for individual functions and data areas using display machine data. Access levels 0 to 3 can also be entered.

Default setting: This machine data is predefined with access level "7". This means that access is granted to all these data areas and functions when the keylock switch is in the 0 position.

Access levels are available for the following system frames:

System frame	MD number	Designation/meaning
ToolFrame	9183	$MM_USER_CLASS_WRITE_TOOLFRAME Protection level write toolholder
PartFrame	9184	$MM_USER_CLASS_WRITE_PARTFRAME Protection level write tool reference point
WPFrame	9185	$MM_USER_CLASS_WRITE_WPFRAME Protection level write workpiece reference point
CYCFrame	9186	$MM_USER_CLASS_WRITE_CYCFRAME Protection level write cycle frame
TraFrame	9187	$MM_USER_CLASS_WRITE_TRAFRAME Protection level write transformation frame
ExtFrame	9188	$MM_USER_CLASS_WRITE_EXTFRAME Protection level write external WO
SetFrame	9210	$MM_USER_CLASS_WRITE_ZOA Protection level write settable WO

Function

The behavior of the **Scratching** and **Set actual value, PRESET** functions is influenced by a number of machine data. The selected settings affect operation (how softkeys and values are displayed, and how the entered values are stored).

With the introduction of system frames, two variants are available for the functions. The variants are differentiated by a channel-specific machine data:

MD 28082: $MC_SYSTEM_FRAME_MASK		
Bit:	0	Actual value setting, scratching
	1	External work offset
	2	TCARR, PAROT
	3	TOROT, TOFRAME

MD 28082: $MC_SYSTEM_FRAME_MASK		
	4	Workpiece reference points
	5	Cycles
	6	Transformations

Version 1 without system frame:

The MD does not exist or Bit 0 for the system frame for scratching or actual value setting is not set.

Version 2 with system frame:

Bit 0 for the system frame for scratching and actual value setting is set in the MD.

Display machine data MD 9422: $MM_MA_PRESET_MODE defines the PRESET/basic offset function in JOG mode.

0: No softkey

1: PRESET in the machine operating area (default setting)

2: Actual value setting

3: Actual value setting (see online help)

Note

To hide the PRESET, Set actual value, and Scratching softkeys, MD 9220: $MM_USER_CLASS_PRESET can also be set.

Actual value setting

- **Actual value setting with system frame:**

  G500 active: The values are written to the system frame.

  G500 not active: An error message is output.

- **Actual value setting without system frame:**

  G500 active (adjustable frames reset): Values are written to the first basic frame according to the entry in display MD 9245: $MM_MA_PRESET_FRAMEIDX = Index of basic frame

  G500 not active: An error message is output.

Scratching

When this is selected the active frame is offered. A different frame can be selected by means of an entry in the "Work offset" field.

- Scratching with system frame:

  With G500 in the "Work offset" field, the values are written to the system frame.

  With a value other than G500 in the "Work offset" field, the values are written to the frame defined in the field.

  You **cannot** choose any tool other than the one in the spindle. If there is no tool in the spindle, you can choose any tool you want.

- Scratching without system frame:

  With G500 in the "Work offset" field, the values are written to the first basic frame.

  Values are written to the specified basic frame according to the entry in display MD 9245:
  $MM_MA_PRESET_FRAMEIDX = Index of basic frame

  With a value other than G500 in the "Work offset" field, the values are written to the frame defined in the field. You can select any tool you want.

See also:

Operation on HMI: Operating Manual HMI Advanced

3.2.2 Fine work offset and base offset

Function

This function refers to the offset in the "Settable work offset" and "Basic work offset" screen forms. Another column for the fine offset is inserted alongside the column for entering the offsets. The inputs are checked against the display machine data.

MD 9203: USER_CLASS_WRITE_FINE	Access level for protection
MD 9451: MM_WRITE_ZOA_FINE_LIMIT	Absolute value of the change

Actual value display: settable zero system

Via MD 9424: MA_COORDINATE_SYSTEM can be used to specify how the actual values are displayed:

- Position of the workpiece coordinate system (programmed position, corresponding to default setting) or
- Take-up position of the active tool relative to the workpiece zero.

MD 9424 = 0	Display in workpiece coordinate system, WCS (default)
MD 9424 = 1	Display in the settable zero system, SZS (fixture position of the active tool)

Example:

Program	WCS display	SZS display
`N110 X100`	100	100
`N110 X100`	0	0
`N120 X0`	0	0
`N130 $P_PFRAME=CTRANS(X,10)`	100	110
`N140 X100`		
`N150 ...`		

Note

To set the actual-value display: see the machine manufacturer's information.

3.2.3 Changing the access level for the "Basis WO" softkey

Parameters operating area

MD 9247: $MM_USER_CLASS_BASE_ZERO_OFF_PA can be used to set the access level from which the "Base WO" softkey is displayed in the "Work offset" window in the "Parameters" operating area. At the same time, the basic frames are also displayed or hidden in the "Work offset" window and in the "Active WO + compensations" window.

Machine operating area

MD 9248: $MM_USER_CLASS_BASE_ZERO_OFF_MA can be set to select the access level from which "WO base" softkey is displayed in the "Scratching" screen in the Machine operating area or G500 can be entered in the work offset field.

3.2.4 Redefining access levels for NC data

Function

The preset access levels can be modified in the definition file. For machine data, only lower access levels than the configured access level are permitted; for setting data, higher levels are also permitted. The file becomes active when the next _N_INITIAL_INI is read in. Different access levels are specified for writing or reading (parts program or PLC).

Example:

```
_N_DEF_DIR/_N_SGUD_DEF                         ; File for global variables
                                               ; $PATH=/_N_DEF_DIR
REDEF $MA_CTRLOUT_SEGMENT_NR APR 7 APW 4       MD 30100
                                               ; (APR ... Read access)
```

```
REDEF $MA_ENC_SEGMENT_NR APR 2 APW 2            MD 30210
                                                ; (APW ... Write access)
REDEF $SN_JOG_CONT_MODE_LEVELTRIGGRD APR 2 APW 2
M30                                             SD 41050
```

MD 30100: $MA_CTRLOUT_SEGMENT_NR has access level 2/7, i.e. access level 2 is required for write access (corresponding to password) and access level 7 for read access.

In order to access the machine data area, key-operated switch setting 3 or higher is required.

Undoing the access level change

In order to undo access level changes, the original values have to be restored.

Example:

```
_N_DEF_DIR/__N_SGUD_DEF                         ; File for global variables
                                                ; $PATH=/_N_DEF_DIR
REDEF $MA_CTRLOUT_SEGMENT_NR APR 7 APW 2
                                                ; (APR ... Read access)
REDEF $MA_ENC_SEGMENT_NR APR 0 APW 0
                                                ; (APW ... Write access)
REDEF $SN_JOG_CONT_MODE_LEVELTRIGGRD APR 7 APW 7
M30
```

Definition files for REDEF

For programming REDEF commands - the same as for the GUD definitions - separate definition files exist, which are evaluated when then control boots.

/_N_DEF_DIR/_N_UACCESS_DEF	// Definition file for access level instructions of the end user
/_N_DEF_DIR/_N_MACCESS_DEF	// Definition file for access level instructions of the manufacturer
/_N_DEF_DIR/_N_SACCESS_DEF	// Definition file for access level instructions for Siemens system applications (e.g. standard cycles, ShopMill, etc.)

A additional Siemens definition file, which is used to configure the system, is saved on the PCMCIA card or CF card in the NC system software.

Note

Compatibility

If the access to machine data that was authorized by the command "REDEF <machine data> APR <access level> APW <access level>" is declined with alarm 15420, you must proceed as follows:

When the function is activated or when the system is upgraded, the REDEF commands from existing GUD definition files must be imported into the new definition files _N_SACCESS_DEF, _N_MACCESS_DEF, _N_UACCESS_DEF.

Access levels for NC commands

Certain NC commands can be linked to access levels to restrict their execution to authorized personnel only.

If no corresponding execution authorization exists, the processing of the part program is aborted with alarm 14018. The preset for the current execution authorization corresponds to the access authorization on the control unit, i.e. key-operated switch setting 0 to 3 or a password for end users as far as Siemens.

References: Programming Manual, Chapter "Access levels for NC commands".

3.2.5 Creating plain text for PLC machine data

Overview

Application-specific and language-dependent plain text can be configured depending on the version [Index] for PLC machine data.

The text files with the name **NCTEA_xx.TXT** (xx = language code GR, UK, ...) can be created in the following directories:

user\language,

add_on\language

oem\language

Entry in the following format:

<Number of entries> ; number of entries

<MD number>**[**<Index>**]** <Text> ; Number[Index] text

Example

14510[i] User data (INT)	i = 0 ... 255
14514[i] User data (FLOAT)	i = 0 ... 31
14516[i] User data (HEX)	i = 0 ... 255

Entry in the file NCTEA_GR.TXT:

```
3
14510[0] My integer text for PLC-MD 14510.0
14514[0] Text for float data 14514.0
14516[2] This is the text in HEX for MD14516.2
```

result in the display:

```
14510[0] My integer text for PLC-MD 14510.0
14514[0] Text for float data 14514.0
14516[2] This is the text in HEX for MD14516.2
```

Note

PLC machine data texts are created and handled in the same way as alarm texts.

Language code for the text files according to the Table: "Languages and ANSI tables / code page used".

See also

Classifying user alarms (Page 83)

3.2.6 Set tool offset to be active immediately

Function

Wit the display MD 9440: ACTIVATE_SEL_USER_DATA can be used to specify that changes in the tool offsets are activated immediately even if the NC channel is not in "Reset" or "Stop" state. This function is always active.

⚠ CAUTION

The offset is applied after NC Start of the Reset in response to the next programmed axis movement in the part program.

References: Function Manual Basic Functions: Axes, coordinate systems, frames (K2)

3.2.7 Hiding axes

Function

Via the channel-specific MD 20098: DISPLAY_AXIS to suppress the display of the axes in the Machine area. All axes are displayed by default.

Axes can be hidden in the following windows in the Machine operating area.

- Small axis window (5 axes, default setting for machine)
- Wide axis window (5 axes with feed and override)
- Large axis window (8 axes, large font)

In this way you can specify separately for the view in the machine or workpiece coordinate system whether or not the axis is displayed:

Bits 0 to 15: Display (=1)/hide (=0) geometry axes (WCS)

Bits 16 to 31: Display (=1)/hide (=0) machine axes (MCS)

Reference point approach and Safety Integrated (special cases)

Machine data 20098: DISPLAY_AXIS is not used for the displays in the referencing operating mode, namely the "Axes Reference Point and Acknowledge Safe Position" display. All machine axes are always displayed.

The operator cannot change the values for the basic work offset without the appropriate access authorization.

Access authorization 7 is entered as the default value in MD 9247 and MD 9248.

20098	DISPLAY_AXIS			
MD number	Display or hide axis on the HMI			
Default setting: 0xFFFFFFFF			Min. input limit: 0	Maximum input limit: 0xFFFFFFFF
Changes effective after POWER ON			Protection level: 2/7	Units: -
Data type: DWORD			Applies from SW version: 4.4	
Significance:	ID as to whether the axis should be displayed as machine, geometry or help axis.			
Bits 0 to 15: WCS	Bit 0	= 1 = 0	Display geometry axis in actual-value windows Hide geometry axis in actual-value windows	
	Bit 1	= 1 = 0	Display geometry axis in reference point windows Hide geometry axis in the reference point windows	
	Bit 2	= 1 = 0	Display geometry axis in n preset / basis offset / scratching window Hide geometry axis in preset / basis offset / scratching window	
	Bit 3	= 1 = 0	Display geometry axis in handwheel selection window Hide geometry axis in handwheel selection window	

20098	DISPLAY_AXIS		
Bits 16 to 31: MCS	Bit 16	= 1 = 0	Display machine axis in actual-value windows Hide the machine axis in the actual-value windows.
	Bit 17		Not assigned
	Bit 18	= 1 = 0	Display machine axis in the base offset window Hide machine axis in base offset window
	Bit 19	= 1 = 0	Display machine axis in the handwheel selection window Hide machine axis in the handwheel selection window

3.2.8 Memory for cycles in the DRAM

Overview

Cycles can be stored in the DRAM of the NCU. This makes more SRAM memory available to the user for parts programs and workpieces.

Function

Under "Start-up" → "NC" → "NC Memory", the user interface for HMI-Advanced offers users with corresponding access authorizations the possibility of determining the utilization of DRAM and to adjust its size via MD 18351: MM_DRAM_FILE_MEM_SIZE. A change to the DRAM memory causes a complete reorganization of the NCK memory.

NOTICE
You must save all relevant data before changing the memory sizes. If machine data is changed then this results in a re-organization of the NCK memory.

Cycles saved in the DRAM

Via MD 11290: DRAM_FILESYSTEM_MASK can be used to set which objects should be loaded in the DRAM:

Bit 0-n	= 0:	The data for the specified directory is to be stored in the SRAM.
	= 1:	The data for the specified directory is to be stored in the DRAM. Depending on MD 11291: DRAM_FILESYST_SAVE_MASK a copy is kept in the FFS on the NC card.
Bit 0		CST directory (Siemens cycles)
Bit 1		CMA directory (machine manufacturer cycles)
Bit 2		CUS directory (user cycles)

Saving in the Flash File System (FFS) (powerline only)

MD 11291: DRAM_FILESYST_SAVE_MASK can be used to set which objects should be saved in the DRAM on the FFS:

Bit 0-n	= 0:	No files are saved. If the control is switched off, the data on the NCK is lost.
	= 1:	The data in the DRAM is saved in the FFS of the NC card.
Bit 0		CST directory (Siemens cycles)
Bit 1		CMA directory (machine manufacturer cycles)
Bit 2		CUS directory (user cycles)

Saving in the FFS avoids having to reload the objects when booting.

NOTICE

Direct changes to objects stored in the DRAM and saved in the FFS are not enabled. Only cycles for saving can be specified in the FFS. Any changes that are required must be made before loading onto HMI-Advanced.

Direct changes to objects in the DRAM that are not saved on the FFS are lost on Power Off.

3.2.9 Specifying how STAT and TU values should be displayed

Singularities in special kinematics and robots

In the Cartesian point-to-point method (PTP) in workpiece coordinates, the position of the machine axes is ambiguous in some places. In order to define these singularities without ambiguity, the machine position STAT and the axes position TU are defined in addition to the workpiece coordinates for this type of transformation (e.g. 5-axis transformation).

Setting the number base

The STAT and TU values are displayed in the axis screens (actual-value window, broad actual-value window with feed and zoom display) and entered in the MDA editor during the teach-in procedure. Whether the values in STAT and TU are displayed in binary, decimal, or hexadecimal format is specified via the display machine data:

MD 9242: MA_STAT_DISPLAY_BASE	Numerical basis for display of moving joint STAT
MD 9243: MA_TU_DISPLAY_BASE	Numerical basis for display of rotary axis position TU

Possible values are:

02	Display as binary value
10	Display as decimal value
16	Display as hexadecimal value

These settings apply to both the actual-value window and the editor window.

Example

In the NC program, the movement of axes from X120 Y20 Z-50 to X200 Y200 Z-120 looks like this in binary display:

```
X=120 Y=20 Z=-50 STAT='B010' TU'B011101'
X=200 Y200 Z=-120 STAT'B110' TU='B011111'
```

3.2.10 Inverting the spindle icon of the spindle display

Function

The display of the spindle icon can be inverted using the channel-specific machine data MD 9033: MA_DISPL_INVERT_DIR_SPIND_M3: Normal or inverted display is possible. For example, if you want the same icon to be displayed at M4 for the counter-spindle as at M3 for the main spindle, set bit[n]=1 (see inverting M3/M4 at the control interface).

9033	MA_DISPL_INVERT_DIR_SPIND_M3	
MD number	Spindle rotation direction display	
Default value: 0x0000	Min. input value: 0x0000	Max. input value: 0x7FFFFFFF
Change effective: IMMEDIATE	Protection level (R/W) 3/4	Units: -
Data type: LONG	Applies from SW version: 6.2	
Significance	Spindle rotation direction display: Bit[n]=0 M3 is displayed as clockwise rotation in the icon Bit[n]=1 M3 displayed as clockwise rotation in the icon n = spindle number – 1 (i.e. bit 0 is spindle S1) Note: For an appropriately set MD 9014: USE_CHANNEL_DISPLAY_DATA MD 9033 can be managed channel-specifically in the NC.	

3.2.11 Representing the spindle utilization

Function

Up to now, the display range of the spindle utilization in the Machine main screen was represented at 100%. You can set the display range of the bar graph for representation of the spindle load to 200% using display MD 9429: MA_SPIND_POWER_RANGE.

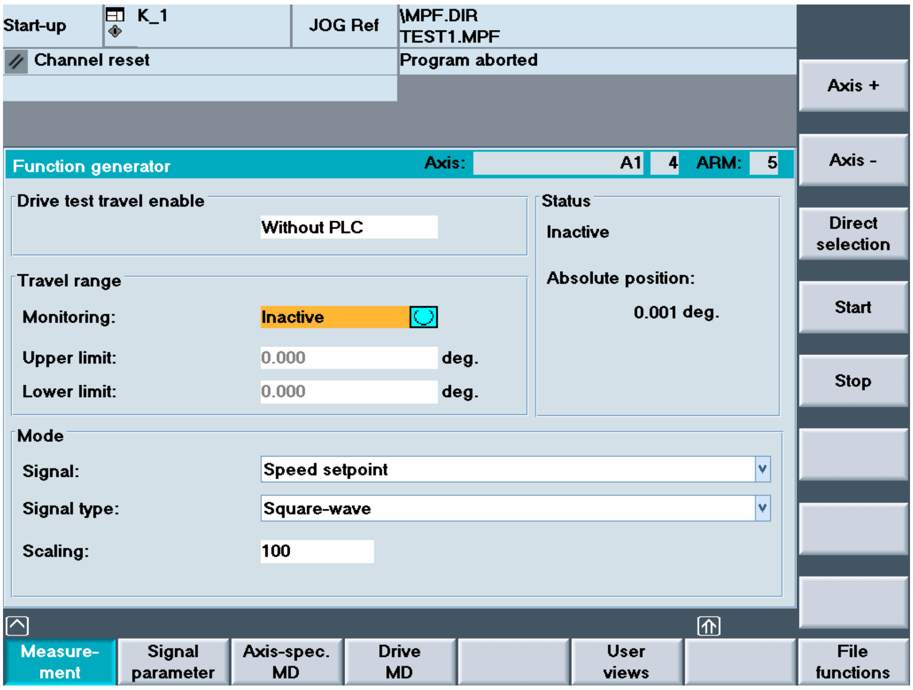
9428	$MM_MA_SPIND_MAX_POWER		
MD number	Maximum value of the spindle power display		
Default setting: 100		Min. input limit: 100	Max. input value: ****
Changes effective after: POWER ON		Protection level: 3/4	Units: %
Data type: WORD			Applies from SW version: 6.4
Significance	In this MD, enter the factor by which the supplied spindle utilization will be multiplied.		

9429	$MM_MA_SPIND_POWER_RANGE		
MD number	Display range for spindle utilization		
Default setting: 100		Min. input limit: 100	Max. input value: ****
Changes effective after: POWER ON		Protection level: 3/4	Units: %
Data type: WORD			Applies from SW version: 6.4
Significance	In this MD, you specify the display range of the bar graph for the spindle utilization display. Depending on the value entered, the percentage values displayed and the extent of the color areas change. **Value 100:**Percentage values 0%, 80%, and 100% are displayed. The color display changes from green to red starting at 80% if a value of 100 is also set in display MD 9428. **Value > 100, e.g. 200:**Percentage values 0%, 200%, and 100% are displayed. The color display changes from green to red starting at 100%. If you set the value to 200, you must enter the normalization factor in display MD $MM_MA_SPIND_MAX_POWER.		

Determining the normalization factor

The normalization factor is determined by measuring the spindle using a function generator, servo trace function, and information in the motor data sheet.

Function generator



Start-up K_1 JOG Ref \MPF.DIR TEST1.MPF

Channel reset Program aborted

Function generator Axis: A1 4 ARM: 5

Drive test travel enable: Without PLC

Status: Inactive

Absolute position: 0.001 deg.

Travel range

Monitoring: Inactive

Upper limit: 0.000 deg.

Lower limit: 0.000 deg.

Mode

Signal: Speed setpoint

Signal type: Square-wave

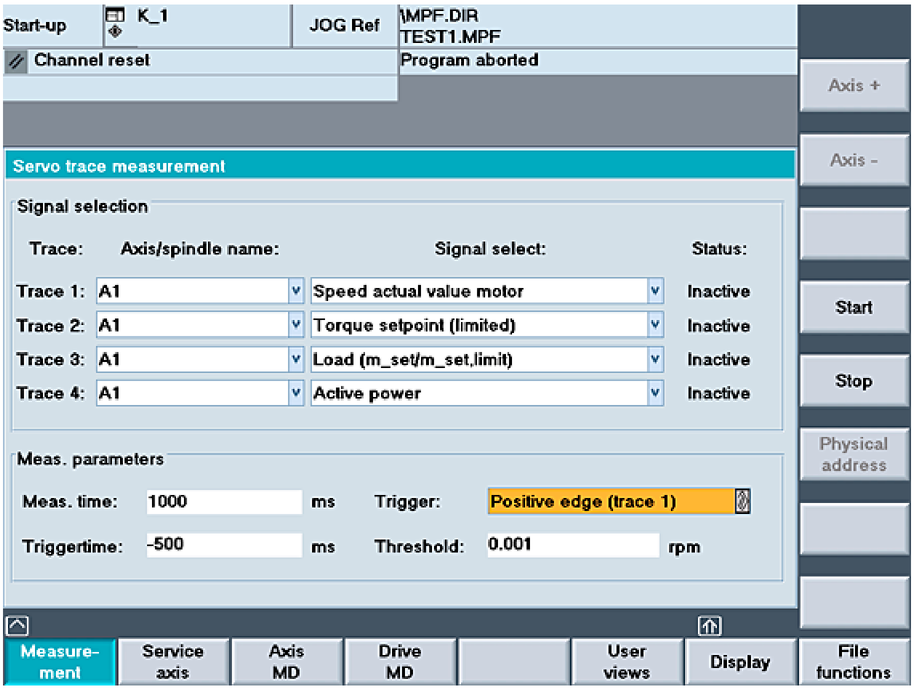
Scaling: 100

Figure 3-4 Function generator

1. Open the following window in the "Start-up" → "Optimization/Test" → "Function Generator" operating area:
2. In the screen form that appears, select a signal, a signal type and a factor for normalization under "Mode".
3. Continue by pressing the "Signal parameters" softkey. The "Function generator parameters" screen form is displayed.
4. Enter the settings for amplitude, cycle duration, pulse width, and limitation. Refer to the motor data sheet to obtain this data.

NOTICE
If the function generator and measuring function are used on virtual axes, this leads to an abort by the NCK.

Servo trace function



Start-up | K_1 | JOG Ref | \MPF.DIR TEST1.MPF

Channel reset | Program aborted

Servo trace measurement

Signal selection

Trace:	Axis/spindle name:	Signal select:	Status:
Trace 1:	A1	Speed actual value motor	Inactive
Trace 2:	A1	Torque setpoint (limited)	Inactive
Trace 3:	A1	Load (m_set/m_set,limit)	Inactive
Trace 4:	A1	Active power	Inactive

Meas. parameters

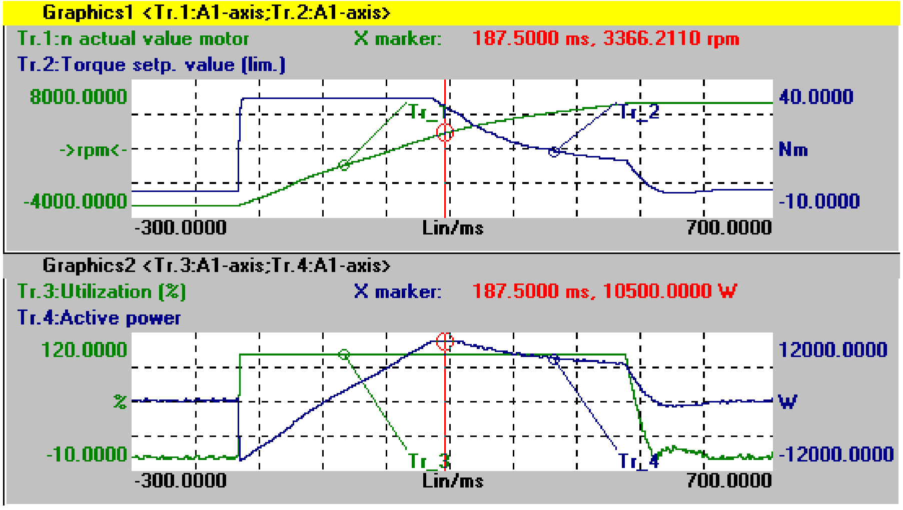
Meas. time: 1000 ms   Trigger: Positive edge (trace 1)

Triggertime: -500 ms   Threshold: 0.001 rpm

Figure 3-5   Servo trace: Main screen

1. Open the Trace function main screen by pressing the "Drive/servo" softkey followed by the "Servo trace" softkey in the "Start-up" operating area.
2. Select the variables to be measured from the drop-down list in the "Signal selection" area. Refer to the motor data sheet to obtain the settings for the measuring period and trigger time. After parameterization, start the measurement by pressing the "Start" softkey.
3. Select the constant power range or constant torque range in order to calculate the normalization factor using these data.

Example: In the diagram, the measured power is 10.5 kW.



Graphics1 <Tr.1:A1-axis;Tr.2:A1-axis>

Tr.1:n actual value motor X marker: 187.5000 ms, 3366.2110 rpm

Tr.2:Torque setp. value (lim.)

Graphics2 <Tr.3:A1-axis;Tr.4:A1-axis>

Tr.3:Utilization (%) X marker: 187.5000 ms, 10500.0000 W

Tr.4:Active power

Figure 3-6 Servo trace: Evaluation

If the utilization display is to indicate 100% for nominal data and higher values at operation above the nominal working point, the normalization factor setting can be calculated from the motor data or from the diagram and measurement as follows:

MA_SPIND_MAX_POWER = Maximum_of_the_measured_power / rated power_S1 * 100

MA_SPIND_MAX_POWER = 10500W / 7000W * 100 = 150

Enter the factor 150 in the machine data.

Programming PLC Functions 4

4.1 Activating the data transfer between the PLC and NCK

Function

Data transfer from/to NCK with the following properties can be started via the PLC:

- Data transfer from the hard disk to or from the NCK (workpiece directory, part program, etc.) using **job lists**.
- The "Job list" function is used in HMI-Advanced to process part programs - e.g. loading, unloading, selecting or executing from the hard disk.

By means of the interface between the PLC and HMI in DB19, the PLC issues jobs to the HMI that initiate a data transfer between the HMI (PCU hard disk) and NCK.

Specifying the user control file: DB19.DBB16 (PLC → HMI)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Always = 1	PLC index for the user control file ; value: 1 - 127						

Specifying the job list: DB19.DBB17 (PLC → HMI)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLC line offset in the user control file; value: 1 - 255							

Job byte of PLC: DB19.DBB13 (PLC → HMI)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Selection	Load	Unload					

Acknowledgment byte from HMI for the current data transfer status:
DB19.DBB26 (HMI → PLC)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Selection	Load	Unload		Active	Error	OK	res. PLC

NOTICE

Machine manufacturer

See machine manufacturer's specifications.

References:

- Operating Manual HMI-Advanced
- Function Manual Basic Functions: Various NC/PLC interface signals and functions (A2)

Job processing

A PLC job is processed according to the following scheme:

- The PLC may only initiate a job in the job byte if the acknowledgment byte is 0.
- The HMI mirrors the job (excluding the parameter set) in the acknowledgment byte (signaling to the PLC that its job is being processed). The PLC receives a "job active" signal for an unfinished job.
- Once the operation is finished (without error or with error), the PLC must respond again and delete the job byte.
- The HMI then sets the acknowledgement byte to 0. This ensures a sequential procedure.

User control file

The names of the job lists to be executed are stored in control file PLC_IN_OUT_xxx.TEA in the display machine data directory (/BD.DIR) on the HMI (= PCU hard disk), where "xxx" represents a number between 001 and 127. The file contains a maximum of 127 references to job lists. Users should create the control file.

The PLC sets an index (1 - 127) via DB19.DBB16, which determines the control file in which the job list names are stored:

- **Example: User control file PLC_IN_OUT_003.TEA**

  ```
  N1 /WKS.DIR/OTTO.WPD/OTTO.JOB
  N2 /WKS.DIR/WELLE.WPD/KANAL1.JOB
  N3 ...
  N4 ...
  ```

  The line numbers N... can be omitted. The file can be edited.

- **Example: Loading a job list**
  - Existing file PLC_IN_OUT_003.TEA with the following entries:

    ```
    N1 job list A
    N2 job list B
    N3 job list C
    ```

  - Acknowledgement byte DB19.DBB26=0
  - Defining the PLC index for the control file (DB 19.DBB16=131)
  - Defining the PLC line offset 2 (DB 19.DBB17=2)
  - PLC job: Loading (DB 19.DBB13.6=1)
  - Job list B is loaded
  - HMI signals to the PLC: Job list is loaded (DB19.DBB26.6=1)

Note

This function is not available if the "Machine" operating area is suppressed from the HMI startup.

Error IDs at the PLC

The HMI outputs the following error IDs to the PLC via DB19.DBB27:

Value	Significance
0	No error
1	Invalid number for control file. (value in DB19.DBB16 < 127 or invalid)
2	Unable to read DB19.DBB15
3	Control file /BD.DIR/PLC_IN_OUT_xxx.TEA not found. (value in DB19.DBB16 invalid)
4	Invalid index in control file. (incorrect value in DB19.DBB17)
5	Selected job list in control file could not be opened.
6	Error in job list (job list interpreter returns error)
7	Job list interpreter returns empty job list
9	Error while executing job list

Conflict resolution

When a job list is executed manually (via operator input), prompts can occur in the protocol, e.g. if an MPF file is located both on the NC as well as on a hard disk and the two MPF files are different.

The default setting is "yes," i.e., all prompts are responded to with Yes. If "no", all prompts are responded to with No. The default setting for the response is "yes", i.e., all prompts are responded to with Yes. If "no", all prompts are responded to with No. The response to the prompt is set in the file MASCHINE.INI:

```
[PLC_REQUEST]
DefaultAnswer=yes
```

Multiple operator panel fronts/NCUs M:N (only SINUMERIK powerline)

In an m:n system, HMI-Advanced is assigned to a netmaster NCU/PLC in NETNAMES.INI. The bus address of the NCU is configured in the [param network] section.

```
e.g.: [param network]
 netmaster=NCU_LINKS
```

This setting for the netmaster NCU/PLC means that data transfer is only activated by the NCU/PLC with the symbolic name NCU_LEFT.

4.2 Transferring the actual task number of the HMI to the PLC

Task number

A task describes the position at which an operating area is attached in the HMI area menu, i.e., the highest level.

Softkey bars

The following fixed relationship exists between softkey bars of the area menu and tasks:

1. Softkey bar:

Softkey	1	2	3	4	5	6	7	8
Task number	0	1	2	3	4	5	6	7

2. Softkey bar:

Softkey	1	2	3	4	5	6	7	8
Task number	8	9	10	11	12	13	14	15

3. Softkey bar:

Softkey	1	2	3	4	5	6	7	8
Task number	16	17	18	19	20	21	22	23

Information on which application is assigned to which task number is stored/configured in the [TaskConfiguration] section in the REGIE.INI file.

A fourth softkey bar can be used for OEM applications:

Softkey	1	2	3	4	5	6	7	8
Task number	96	97	98	99	100	101	102	103

Task numbers 24 to 95 are already reserved for child tasks of standard applications or for specific OEM tasks. The associated label texts can be entered in the language-dependent re_xx.ini files.

Application

For manufacturer-configured user interfaces, proper concurrence of key strokes on the machine control panel and the displayed operating areas/screens is essential.

Example:

The "Move unit" function is selected in a screenform. On the MCP are the **"forward" and "backward" traversing keys**, which are used to move the unit in real time.

To prevent a new screen from being displayed before the traversing key movement has been completed, for example due to an operating area switchover in the HMI, the PLC is always able to know and check the current task number.

Configuration

The current task number is transferred to the PLC in a data block section that can be specified as a string in the display MD 9032: HMI_MONITOR as follows:

`DBxx.DBByy` with:

xx is the number of a data block

yy is the byte number of the 1st byte of the transfer area (word limit)

The transfer area is 8 bytes long, the transfer is made in the 1st byte. Bytes 2 to 8 are reserved.

Note

Check that the specified area is not already occupied by input bytes, output bytes or flag bytes. The block number and the byte area are not checked.

Updating

The task number in the first byte is updated for:

- At every task change by the HMI
- When the connection is changed to a different NC in M:N

For the duration of indeterminate states (e.g. during changeover to another NC in M:N), special value 255 is transferred as the task number.

Range of values

The task number can assume values from 0 to 95, including the operating areas configured by the user.

As a result, all visible tasks of the main menu (0 to 31) and the hidden child tasks (32 to 95) are displayed.

4.3 Channel/spindle selection via the PLC

Channel selection

Channel selection or display is initiated as follows (display in "Channel name" field):

```
DB 19, DBB 32 = 01000001 [binary] channel selection
DB 19, DBB 33 = <channel number>
FF for next channel
```

Spindle selection

Spindle selection is initiated as follows (display in spindle window):

```
DB 19, DBB 32 = 01000010 [binary] spindle selection
DB 19, DBB 33 = <spindle number> spindle number to be displayed
```

Error codes

The following error codes, which provide information to the HMI about errors during the function request of the PLC, are stored in DB 19.DBB 36:

Value	Significance
0	No error
1	No function number (DBB32 bit 0-5)
2	Invalid parameter
3	Error when writing to HMI-internal variable
10	Channel not present (DBB33)

Acknowledgement mechanism between PLC ↔ HMI

Via DB19.DBB32 bits 6, 7, there is an acknowledgment mechanism that ensures that the HMI and the PLC coordinate the channel selection:

The PLC sets Bit 6 to 1 = "Function request". The function request can only be set by the PLC if bits 6 and 7 have the value zero.

After displaying the text in the header the HMI writes zero again in bit 6 and the interface is enabled for the PLC for the next action.

Note

If several HMI units are assigned to one NCU/PLC unit (M:N), the PLC must first determine from the control interface which of the HMI interfaces is presently active. Channel/spindle displays can only be requested by the PLC for the active HMI unit. The interface for selecting channel/spindle displays for the 2nd HMI unit is located in DB19, DBB82-86. The use is the same as described for DBB32-36.

4.4 Configuring the display of messages in the header

Function

Using the PLC (also see the Chapter Cross-channel status display with icons), it is possible to configure a two-line text display to appear in the "Program name" field in the header of JOG and MDA modes (e.g. selected slide, activated handwheel).

Syntax:

The message texts can consist of:

- A language-specific part (from alarm/message text file *.COM)
- Three parameters (%1, %2, %3)

The maximum length of the texts to be displayed is 30 characters per line.

User PLC-DB

The display must be configured via a user PLC data block.

The number of the DBx with DBBy must be entered by the user in the HEADER.INI file as follows:

```
[UserTextInfo]
USER_TEXT_BASE=DBx.DBBy
```

with x data block number, y starting byte in the block

If this interface is specified, the text length limiting must also be activated.

```
Textlength = 33
```

The name of the PLC alarm text file (e.g. "alpu_") should be entered by the user in the MBDDE.INI file in the [TextFiles] section under "UserPLC":

```
[TextFiles]
UserPLC=alpu_
```

User DB interface to PLC

Two lines in the header can be used for the display:

- Program path line
- Program name line

The interface of the user DB provides a start data byte (job ID by the HMI) and subsequent bytes with the objective that an axis index and a text (length, 12 characters/special characters) can be output.

If "00" is written to a byte during text input, the HMI interprets this as the end of text. Entries after this "00" entry are not displayed.

Table 4- 1 User PLC-DB interface

Program path line	Start DBBx	Start-DBWx +1	Start-DBBx + 3	Start-DBBx + 4	Start-DBBx +5-17
Data type	Byte	Word	Byte	Byte	String
Significance	Bit0=1: there is something to do All bytes=1: job is acknowledged by HMI All bytes=0: PLC can write to interface again	Offset for text to be displayed from alpu_xx.com, 700000 is the basis address and is always added internally =0: Delete line	Number from 0 ...255 (corresponds to the 1st parameter %1)	Index from –127 ...128 negative number (-1=0xFF, -2=0xFE) is the machine axis index, the positive number is the channel axis index. The index is then converted to an axis name before being displayed. (corresponds to the 2nd parameter %2)	String containing a maximum of 12 characters (corresponds to the 3rd parameter %3)
Program name line	**Start-DBBx +20**	**Start-DBWx +21**	**Start-DBBx +23**	**Start-DBBx +24**	**Start-DBBx +25-37**
Data type	Byte	Word	Byte	Byte	String
Significance	As above	As above	As above	As above	As above

Note

The Start-DBBx may only be supplied for a new job with bit 0 = 1 if all of the bytes after the previous job were again zero and the parameters for the new job Start-DBBx+1 to StartDBBx+5 are correctly set.

Special characters that have a special meaning within the syntax for this function (", %, $, #) must not appear within the user's character string (Start DBB+5…).

Start DBBx	HMI detects a job and acknowledges it (all bytes=0).
Start DBWx+1	Start DBWx+1 contains the offset for a text number with a base of 700,000 (user PLC texts). If this parameter = 0, the corresponding line is deleted.
Start-DBBx +3	Start DBBx+3 contains a number that is displayed without being converted.
Start-DBBx +4	Start DBBx+4 contains an axis index. Negative value = machine axis index (AXCONF_MACHAX_NAME_TAB[Index-1], MD 10000), Positive value = channel axis index (AXCONF_CHANAX_NAME_TAB[Index-1], MD 20080), These indices always refer to the channel currently displayed in the HMI. The index is converted to an axis name before being displayed.
Start-DBB +5	Start DBBx +5−17 contains a character string comprising a maximum of 12 characters.

Structure of PLC program

The PLC program must be laid out in such a way that all parameters are initialized before the first byte is set.

The data for the program name line (second line) are located 20 bytes after the start byte.

Acknowledgement mechanism between HMI ↔ PLC

For the Start DBBx, there is an acknowledgment mechanism that ensures that the HMI and PLC are coordinated:

The PLC sets Bit 0 to 1 "Display user text in header". The function request can only be set by the PLC if Bit 0 has the value zero.

After displaying the text in the header, the HMI writes zero again in bit 0 and the interface is enabled for the PLC for the next action.

Behavior for M: N (only SINUMERIK powerline)

For the function M : N, both text lines for passive switching must be deleted. In active switching, as in area switching, the content of the variables is evaluated and displayed.

Task:

In the header of the JOG and MDA modes, the text "Handwheel acts on axis..." should be output in the field "Program name" in the first line together with the appropriate axis names; the text "slides ... active" should be output with the corresponding slide number in the second line.

Procedure:

1. Identify the PLC ↔ HMI interface in the HEADER.INI file, e.g. DB60.DBB10
2. Enter name of the PLC user text file in MBDDE.INI, e.g.:

   ```
   [TextFiles]
   UserPLC=alpu_
   ```
3. Enter texts in PLC user text file, e.g.:

   ```
   700100 0 0 "Handwheel acts on axis %2"
   700101 0 0 "Slides %1 active"
   ```
4. Initiate text display from the PLC and describe the interface, e.g. for the first display line:

   ```
   DB60.DBW11 = 100
   DB60.DBB14 = 2
   DB60.DBB10 = 1
   ```

   e.g. for the second display line:

   ```
   DB60.DBW31 = 101
   DB60.DBB33 = 1
   DB60.DBB30 = 1
   ```

Result

For an assumed axis configuration of

MD 20080 AXCONF_CHANAX_NAME_TAB[0] = "X"

MD 20080 AXCONF_CHANAX_NAME_TAB[1] = "Y"

MD 20080 AXCONF_CHANAX_NAME_TAB[2] = "Z"

the display is then as follows:

```
"Handwheel acting on Y axis" (first display line)
"Slide 1 active" (second display line)
```

4.5 Starting a block search across several channels

Function

For this block search, the NCK is operated in the program test mode, so that interactions between channel and synchronous actions and between multiple channels are possible within an NCK.

During the "Block search in the program test mode" the NCK outputs all help (auxiliary) functions to the PLC and the part program commands to coordinate the channels (WAITE, WAITM, WAITMC, axis exchange, writing to variables) are executed by the NCK; this means that during this block search, the PLC is brought up-to-date and processing sequences - that comprise the interaction of several channels - are correctly executed within the scope of this block search.

Activating the block search

The function is activated by the "Prog. test contour" softkey. The NCK then starts the block search and switches the selected program in the current channel to "program test" mode:

- The auxiliary functions of the part program are output from the NCK to the PLC.
- No axis movements are executed.

If during the block search the NC finds one of the part program commands WAITE, WAITC or WAITMC, the NCK waits for the specified partner channels, irrespective of the current mode of these partner channels (e.g. normal program execution, program test, search via program test, etc.).

When the NC reaches the specified target block in the current channel, the NC stops the block search and deselects "program test" mode; the auxiliary functions of the target block are no longer output.

A dialog message is output, which must be acknowledged as soon as all the channels involved in the current search have exited "program test" mode.

The message indicates that, depending on the particular part program, REPOS offsets may have been produced during the search operation in the channels involved in the block search and will be retracted by the NC through interpolation the next time it starts. Any REPOS offsets can be approached manually one by one in JOG mode, before program execution is continued at the point in the program reached by the block search by pressing the "NC Start" key.

Block search configurations

Block search in "program test" mode supports the following configurations:

1. Block search in the currently selected channel (the currently selected channel is the channel selected by the HMI).
2. Block search in the currently selected channel and in all channels in which the same workpiece is selected as in the current channel.
3. Block search in the currently selected channel and in all channels that are in the same mode group as the current channel.
4. Block search in all channels of the NCK

Configurations 2 to 4: No block search is performed in the other channels (in contrast to the current channel); "program test" mode is active in the other channels until a stop condition (e.g. WAITMC) is detected.

Setting the search configuration

The various configurations are activated in the MACHINE.INI file.

The following settings are possible:

```
[BlockSearch]
SeruproEnabled=1
; Shows (SeruproEnabled=1) or hides (SeruproEnabled=0) the softkey
used to do a block search in "program test" mode. The softkey
appears in both dialogs "Blocksearch Searchposition" and
"Blocksearch Searchpointer"
SeruproEnabled=0

           =0    Function inactive
           =1    Function active (default setting)
SeruproConfig=1

           =1    Block search in currently selected channel
                 (default setting)
           =2    Block search in currently selected channel and in
                 all channels with the same workpiece
           =3    Block search in currently selected channel and in
                 all channels with the same mode group
           =4    Block search in all channels
```

Changes take effect immediately, i.e., with the next search.

4.6 Cross-channel status display

Function

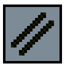
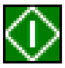
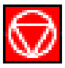
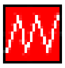
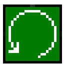
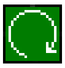
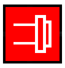
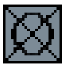
The status of the following components can be output with icons in the status line (in the header):

- Channels
- Spindles
- Machine states

In addition to SINUMERIK icons, manufacturer-defined icons can also be configured in the file HEADER.INI file and in DB19 DBB32-34.

SINUMERIK icons

The following SINUMERIK icons with the TrueColor color setting are available (file name in brackets):

Channel status		Channel RESET	(NC_RESET.BMP)
		Channel active	(NC_START.BMP)
		Channel interrupted	(NC_STOP.BMP)
Feed stop		Feed is not enabled	(FEEDSTOP.BMP)
Spindle status		Spindle is turning counter-clockwise	(SPNDLEFT.BMP)
		Spindle is turning clockwise	(SPNDRGHT.BMP)
		Spindle not enabled	(SPNDSTOP.BMP)
		Spindle STOP	(SPNDM05.BMP)

Storing the SINUMERIK icons

The SINUMERIK icons are stored in

- F:\HMI_ADV\ICONS\640 (for 640x480 screen)
- F:\HMI_ADV\ICONS\800 (for 800x600 screen)
- F:\HMI_ADV\ICONS\1024 (for 1024x768 screen)

The dimensions of the SINUMERIK icons are as follows:

- Resolution 640x480: 16x16 pixels
- Resolution 800x600: 20x20 pixels
- Resolution 1024x768: 27x26 pixels

Note

When using the "Reversible spindles" or "Axis container" functions, the spindle status is output **before** the spindle reversal or assignment to axis/spindle container, i.e. the spindle number displayed under spindle status is the "logical" spindle.

Activating

The function is activated by MD 9052 SHOW_CHANNEL_SPANNING_STATE > 0. Changes in MD 9052 take effect after a POWER ON.

This function is available if it is activated by means of the following machine data element:

9052	SHOW_CHANNEL_SPANNING_STAT		
MD number	Change cross-channel status display		
Default value: 0		Min. input value: 0	Max. input value: 1
Changes effective after POWER ON		Protection level (R/W) 4/1	Units: -
Data type: STRING		Applies from SW version: 6.2	
Significance	Cross-channel status display: 0 = Display previous program status in header 1 = Display in program status line in header according to configuration in Header.ini file.		

SINUMERIK status display

The following **4 predefined displays** are available:

1	Channel status	(active, stopped, RESET) with superimposed spindle stop and feed stop
2	Channel status	(active, stopped, RESET) with superimposed feed stop
3	Channel status	(active, stopped, RESET)
4	Spindle status	Counter-clockwise direction, clockwise direction, spindle stop, spindle halt M05

1 channel status

The channel status, feed stop, and spindle status are output as an icon at the display position. The channel status icons can be superimposed with the "Spindle not enabled" or "Feed not enabled" icon.

The following **hierarchy** applies (from highest to lowest priority):

1. Channel STOP (highest priority)
2. Spindle not enabled
3. Feed is not enabled
4. Channel active
5. Channel RESET (lowest priority)

Syntax:

```
ChanStatFeedStopSpindleStop = <Channel number>
Spindles = <Spindle number1>,...,<Spindle number n>
Channel number: Number of channel
Spindle number: Numbers of the "logical" spindles to be taken into
consideration
(programmed in the DIN code with S1 to Sn; if more than
one applicable entry, the higher number is displayed).
```

2 channel status

The channel status and spindle status are output as an icon at the display position (spindle status can be separately output). The channel status icons can be superimposed with the "Feed not enabled" icon.

The following hierarchy applies (from highest to lowest priority):

1. Channel STOP (highest priority)
2. Feed is not enabled
3. Channel active
4. Channel RESET (lowest priority)

Syntax:

```
ChanStatFeedStop = <Channel number>
Channel number: Number of channel
```

3 channel status

The channel status is output as an icon at the display position. Feed stop is also output in the feed window in the Machine operating area, for example.

Syntax:

```
ChanStat = <Channel number>
Channel number: Number of channel
```

4 spindle status

The spindle status is output as an icon at the display position.

Syntax:

```
SpindStat = <Spindle number> , Chan <Channel number>
Spindle number: Number of (logical) spindle
Channel number: Number of channel optional
```

If the channel number is not specified, then the spindle status for the current channel is displayed. If the channel is changed, the display is updated accordingly.

Configuring in the HEADER.INI

The assignment of the cross-channel status display is defined in the HEADER.INI file. It can optionally be configured with SINUMERIK and user icons.

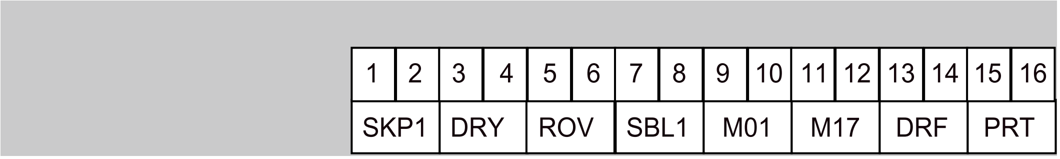
Note

Empty positions need not be specified. Predefined status displays can be disabled in the HEADER.INI file in the OEM or USER directories with the value <empty>, e. g. ChanStatFeedStopSpindleStop = <empty>.

The cross-channel status display and the use of the USER/OEM icons are mutually exclusive; i.e. either "cross-channel status display" or "USER/OEM icons" can be configured. The complete file is to be found in Section HEADER.INI.

Example with 16 display positions:

There are 16 display positions defined in the program status line (header), where each SINUMERIK icon occupies two positions: number + icon.



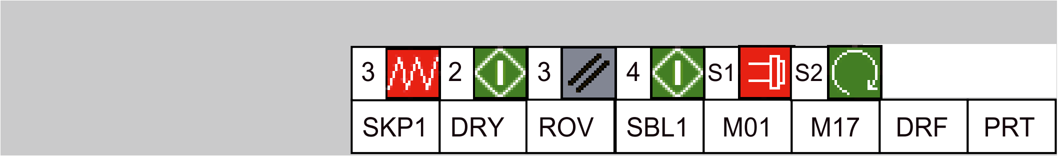
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SKP1		DRY		ROV		SBL1		M01		M17		DRF		PRT	

Figure 4-1 Program status line with 16 positions

The SINUMERIK icons are permanently assigned to an output position. A section is created in the HEADER.INI configuration file for each position used.

Example 1 with SINUMERIK icons:

View of the program status line with SINUMERIK icons for 4 channels and 2 spindles:



3	2	3	4	S1	S2		
SKP1	DRY	ROV	SBL1	M01	M17	DRF	PRT

Figure 4-2 Program status line with icons (1)

These displays always have the predefined channel or spindle number.

Example 2 with SINUMERIK icons:

The status displays for 2 channels and 1 spindle should always be output at the first, third and fifth display position.

HEADER.INI file:

```
[Pos1]                  ;  first display position
ChanStatFeedStopSpindleStop = 3
Spindles = 1,2          ;  Channel status of channel 3 is displayed,
                           superimposed with feed inhibit channel 3
                           and spindle inhibit for spindles1 and 2
[Pos3]                  ;  third display position
ChanStatFeedStop = 2    ;  Channel status of channel 2 is displayed
                           superimposed with the feed inhibit,
                           channel 2
[Pos5]                  ;  fifth display position
SpindStat= 3            ;  spindle status for spindle 3 is displayed
```

The following display, for example, appears in the program status line:

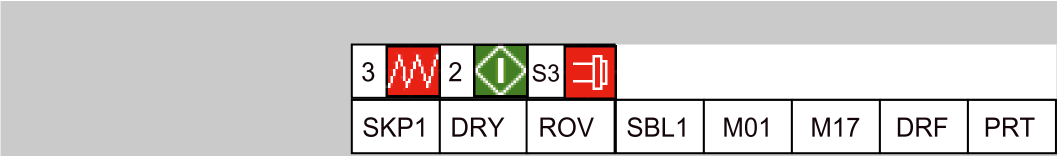


Figure 4-3 Program status line with icons (2)

Deactivating

Icon displays configured in this way can be deactivated if user status displays are used as described in B. Create empty entries in a user-specific HEADER.INI file (not in the hmi_adv directory, because this would be overwritten again when the software is upgraded).

Example:

```
[Pos5]
SpindStat=<EMPTY>
```

See also

HEADER.INI (Page 283)

4.7 User status display (OEM)

Function

Additionally (or alternatively), PLC-controlled (DB x, DBB y) user icons (symbols) linked to specific machine states can be output.

Procedure:

1. Declaration of icons and positions
2. Control of icon call by the PLC

Enter the name of the user icons and the associated positions in the HEADER.INI file in the [UserIcons] section under identifiers UI_0 to UI_31.

```
[UserIcons]
 UI_0= <Icon_00.bmp> , <Position>
 ; UI_0: Identifiers
 ; Icon_00.bmp: Name of user icon (8 characters.bmp)
 ; Position: Display position (1 to 16)
 ...
 UI_31= <Icon_31.bmp> , <Position>
 USER_ICON_BASE = DBx.DBBy
```

`DBx.DBBy`: Double word defined by user for controlling the icon selection. If a bit for a User Icon is set, then the associated UI_x entry should exit, otherwise nothing is displayed. To display an icon according to the above declaration, the PLC sets the corresponding bit in the double word.

Example:

```
Display an icon assigned to UI_0 → bit 0
...
Display an icon assigned to UI_31 → set bit 31.
```

If the bit is reset by the PLC, the associated icon is cleared from the display.

Example:

```
UI_3=Icon_01.bmp,12.
```

If bit 3 is set, the "con_01.bmp" icon is displayed at the 12th position, provided another icon with a higher identifier (>UI_3) and the same position (12) is not active.

Possible applications

With a **unique selection**, a maximum of 16 bits are set simultaneously in the 4 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)
- 2 alternative images for each of the permissible 16 positions, each position requires 2 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:

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 units are assigned to one NC/PLC unit (M:N), the PLC must first determine from the control interface which of the HMI interfaces is active. Icons can be requested by the PLC for the active HMI unit only.

Configuring OEM texts

The machine manufacturer can display his 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.

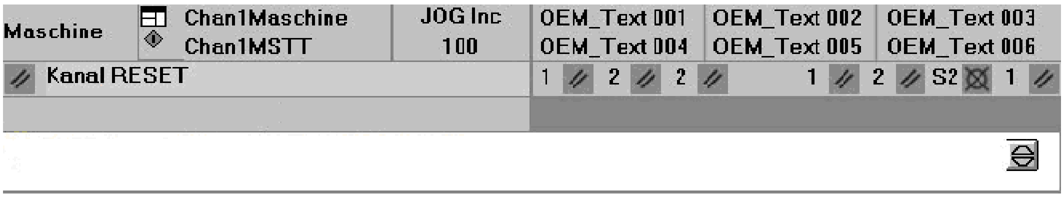


Figure 4-4 Example for OEM text

The data are provided in a user PLC DB. 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 or Start-DBB +18 byte	Bit0=1: there is something to do All bits=1: job is acknowledged by HMI All bits=0: PLC can write to interface again.
Start-DBB +1 word or Start-DBB +19 word	Offset for text from alpu_xx.com to be displayed; 700 000 is the base address and is always summed internally. All bits=0: Delete line
Start-DBB + 3 string or 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 are 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 6 fields are possible).

A hotlink to the first byte for each line is set up in the HMI. If a change occurs in that byte, the HMI reacts. If the HMI has displayed the data, it writes a –1 to the first variable. The HMI is informed about this change by means of the hot link, 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 HMI 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, same as the user icons.

```
[OemIcons]
 OI_0= <name.bmp>, <Position>
 ...
 OI_31= <name.bmp>, <Position>
```

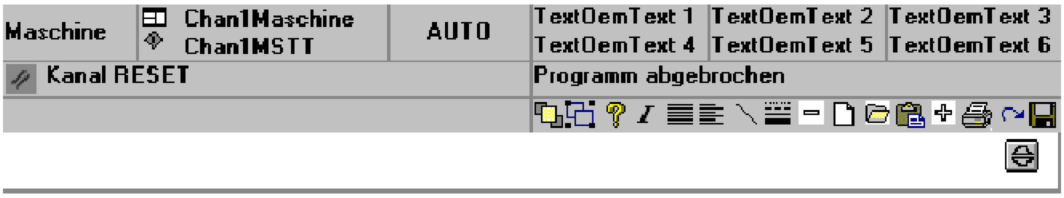
Here, <name> is the file name of the bit map and <Position> is the output position (from 1 to 16) in the display line. Multiple bit maps can be output at the same position. If more than one bit map is active simultaneously at the same position, the bit map with the highest screen number is displayed.

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.



Maschine
Chan1Maschine
Chan1MSTT
AUTO
TextOemText 1
TextOemText 2
TextOemText 3
TextOemText 4
TextOemText 5
TextOemText 6
Kanal RESET
Programm abgebrochen

Figure 4-5 Example for OEM Icons

See also

HEADER.INI (Page 283)

Diagnostics and service 5

5.1 Installing HMI software on a PG/PC

Requirements

Target system:	PG or PC with 600 MHz / 256 MB RAM Available hard disk space: 500 MB
Operating system:	Windows XP Professional
MPI driver:	Version 6
NC connection (SINUMERIK powerline)	CP5511 CP5611
NC connection (SINUMERIK solution line):	Network connection (Ethernet)

"Small fonts" must be selected under "Control Panel → Display, Settings".

Software

• Name	HMI-Advanced for PC/PG
• Installation language	German/English
• Distribution medium	CD
• Can be uninstalled	Yes

Operating sequence

1. Insert the CD in the CD drive
2. Start setup.exe
3. Select language
4. Specify the target directory: Any
5. For SINUMERIK powerline, select:
   – Online with communication to NC
   – The system asks whether you wish to install the MPI drivers
   – Local without communication to NC, with NC simulation
6. For SINUMERIK solution line, select:
   – Online with communication to NC
7. The system lists all the entries you have made

8. On confirmation the copy operation starts
9. At the end of the copying operation, you are prompted to restart the PC.

Note

Installing on a PC/PG:

HMI-Advanced can then only be run under a non-administrator user, if this user has write privileges for directory mmc2 of HMI-Advanced. The installation of HMI-Advanced does not provide these privileges.

5.1.1 NCU Connection Wizard

Use

The NCU Connection Wizard is only installed during the installation of HMI-Advanced on a PG/PC. This assists you during the commissioning of the NCK, PLC and drive and during the configuration of interfaces and the cabling.

With the Connection Wizards, connections are possible both via MPI and Ethernet, depending on the control unit. Use an Ethernet connection for SINUMERIK solution line and an MPI connection for SINUMERIK powerline. The type of connection depends on the software installed on the 840Di control unit.

Starting the "NCU Connection Wizard"

The Connection Wizard is started under "Start menu" → "NCU Connection Wizard" or it is automatically started after a message if no connection to the control unit can be established.

Note

Changes that you made using the "NCU Connection Wizard" only take effect after a system restart.

5.1.2 Starting HMI-Advanced in a separate desktop

Overview

Installing HMI-Advanced on a standard PC creates boundary conditions which cause the following effects when other programs are used:

- Windows that cannot be moved or minimized
- Special keyboard and keyboard filter functions
- Overcrowded task bar
- Changes to color scheme

- Focus control in connection with OEM software
- All applications of a standard PC run under one desktop.

How to call-up HMI-Advanced in a separate desktop as remedy is described in the following section.

Remedy

Through the dynamic creation of a separate desktop for the HMI-Advanced software, the features indicated above are limited to this desktop. In addition, up to three desktops can be defined without these restrictions. Various methods can be used to switch between the desktop with HMI-Advanced and the other desktops at any time. Desktop creation is configurable. The solution is only available for PGs/PCs, not for PCU 50.3.

Desktop applications:

- In addition to a standard desktop that is always present, one or more desktops are created dynamically.
- HMI-Advanced is loaded into a dynamically created, standalone desktop.
- All existing desktops are provided with a user interface that allows switching to another desktop at any time.
- The desktop can be configured using the **HMIDesk.ini** file.

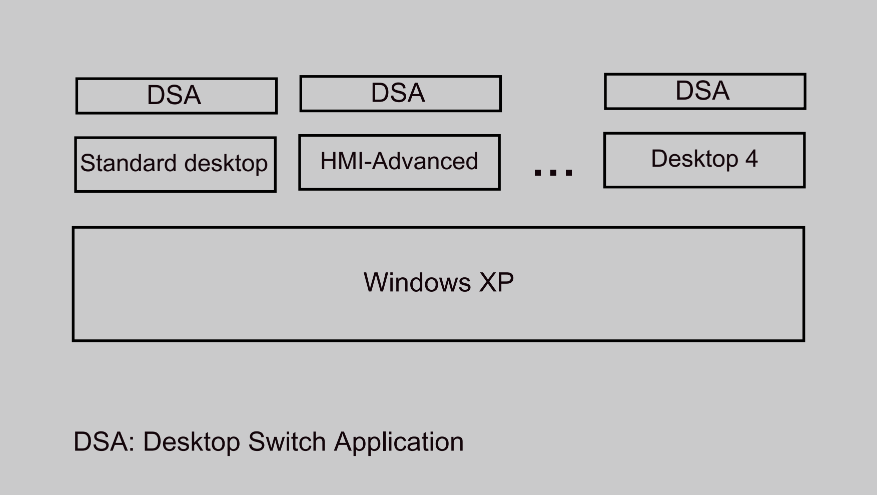


Figure 5-1 Separate desktops for HMI-Advanced and other applications

Note

1. The key combination **CTRL+ALT+DEL** of the task manager only works in the standard desktop. In dynamic desktops, the task manager can only be called directly from the task bar.
2. When a dynamic desktop is closed, the application that is started automatically by inclusion in hmidesk.ini is closed with the desktop. Additional, manually started applications continue to run, however.

   The applications can no longer be used, however, since they are not displayed in other desktops. If the dynamic desktop is started again later, applications that are still running are visible again and can continue to be used.
3. All **external applications** (e.g. ProTool, TRANSLINE) that use the basic HMI-Advanced system must be started on the HMI desktop if they are to be executed in parallel with HMI-Advanced.

   Background: The sequence control subsystems cannot be used beyond desktop limits because Windows restricts window communications to one desktop.
4. **16-bit applications** are always started by Windows in the login desktop and so do not work in a dynamic HMI desktop. The desktop switch can therefore not be used if 16-bit applications are being used in HMI-Advanced. In this case, HMI-Advanced must be started in the conventional way.

5.1.3 Configuring the desktop switch application

Function

The application is configured in the HMIDESK.INI, in the section [Global].

Under[Global], the required number of desktops is specified with **NumberOfDesktops** = n. The 2nd to 4th desktops are created dynamically. Permissible values for n are 2, 3, 4. For each dynamic desktop, additional settings must be specified in [Desktopn].

ControlMode =	Icon	Icons in the task bar (Explorer below)
	Window	Standalone dialog window (WindowOnTop below)
KeyboardControl =	TRUE	Switching can be done using <ALT+1>, <ALT+2>, ... dialog window or task bar.
	FALSE	Switching can be done using the dialog window or task bar.

```
[Desktop n] section
StartupApplication = "Path\Program.exe"
```

An application is started automatically after creating a desktop: On closing the desktop, this application is closed by the desktop switch. Closing of the desktop is synchronized with the closing of the application. During this synchronization, the desktop switch cannot process any other requests.

The synchronization is aborted if the application is still being executed after 15 seconds. In this case, the desktop is closed too.

Note

For the desktop (2), the path and associated program name pointing to the current HMI-Advanced installation are set automatically.

Explorer =	TRUE	Desktop is provided with Explorer and a task bar which contains the icons for the configured desktops.
	FALSE	Desktop is not provided with Explorer or a task bar The desktop can only be selected via the keyboard <ALT+1>, <ALT+2>, ... or via the dialog window.
WindowOnTop =	TRUE	If ControlMode = Window is present, the dialog window of the desktop switch application is constantly displayed.
	FALSE	Dialog window can be covered
CreateDesktop =	Immediately	Start desktop immediately.
	Delayed	Start desktop when selected.
SetHMIColors =	TRUE	With SetHMIColors, the Desktop 2 uses its own color scheme with the advantage that this color scheme does not affect other desktops. This attribute is intended exclusively for use with HMI-Advanced, because it uses its own color scheme which is partially very different from standard Windows.
	FALSE	The desktops are displayed with the standard Windows color scheme, as is the desktop with HMI-Advanced.

When switching between desktops, the desktop switch sets the appropriate color scheme automatically.

Desktop 1 is the user's standard desktop, which is set up automatically by the operating system after logon. For the standard desktop, only the property WindowsOnTop = True can be configured.

- Activating

  Start HMIDesk.exe in the installation directory for HMI-Advanced. The configured desktops are created. HMI-Advanced is started automatically in desktop 2 as the startup application.

- Behavior when creating a desktop

  An active desktop is created with the task bar. The desktop switch window is hidden. The desktop switch is operated via the task bar.

5.1.4 Operating the desktop switch application

Overview

The user interface for the desktop switch application can be configured:

- Keyboard hotkeys
- Dialog window
- Icons in the task bar

Hotkeys:

<Alt+1>	Switches to desktop 1 (standard user desktop)
<Alt+2>	Switches to desktop 2 (HMI-Advanced desktop)
<Alt+3>	Switches to desktop 3 (optional additional user desktop)
<Alt+4>	Switches to desktop 4 (optional additional user desktop)

Starting HMI_Advanced in a separate desktop:

1. First start a second desktop, then HMI-Advanced:

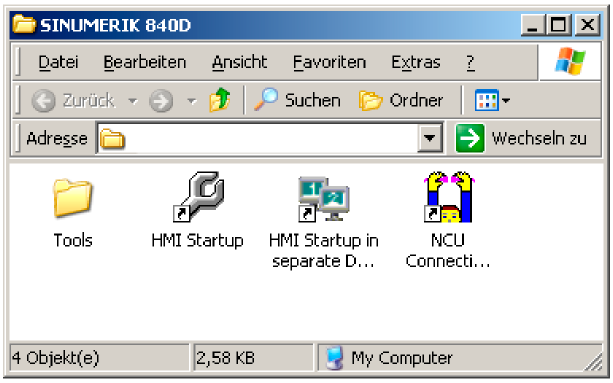
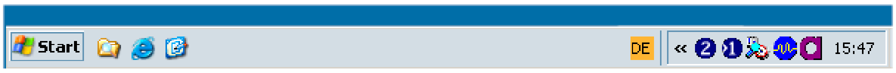


Figure 5-2 Linking to the desktop switching application

2. Switching by clicking on the corresponding desktop number:

1	Standard desktop
2	Desktop with HMI-Advanced
(3, 4)	Other optional desktops



↑ Displaying the desktop

↑ Changing over the active desktop

Note

If WindowOnTop = False is configured in HMIDESK.INI, the window must be brought to the foreground using <Alt + TAB>.

5.2 Saving a screenshot

Application

To provide more information for troubleshooting and problem analysis, you can save a screenshot of a file.

File type and file name for screenshots

You can generate a screenshot with key combination <Ctrl> + <P>. The entire screen is copied to a file with automatically generated file name according to the following scheme:

- scf00001.bmp
- scf00002.bmp
- scf00003.bmp
- etc.

A maximum of 99 screenshots can be stored. After that, the files are overwritten again starting with scf00001.bmp.

The screenshots are saved as BMP files to the temporary directory F:\ALTMP of the PCU (entry in MMC.INI):

```
[DIRECTORIES]
TempDir=F:\TMP
AlDir=F:\ALTMP
```

5.3 Displaying versions

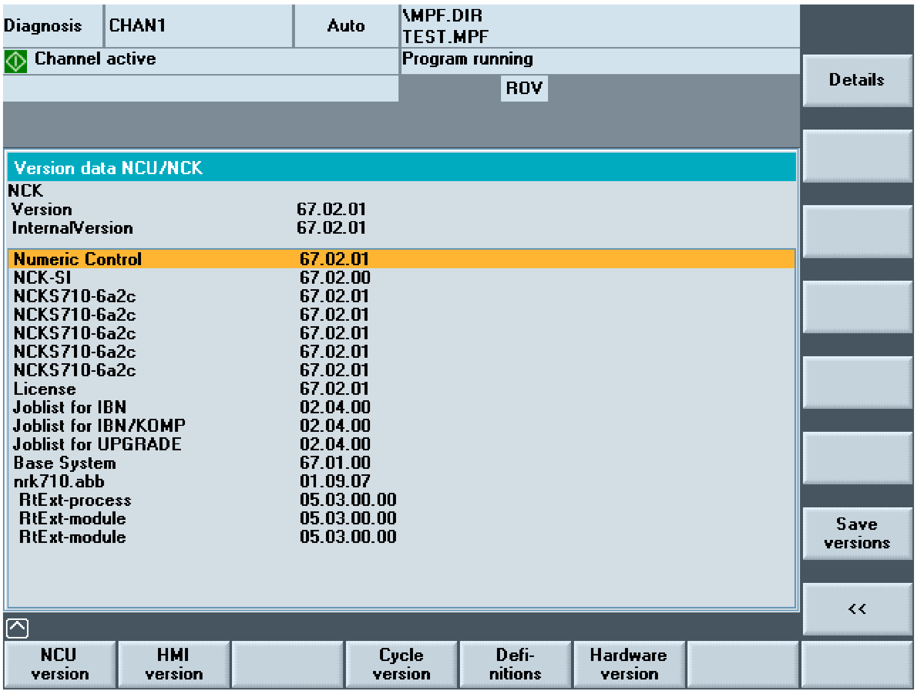
Overview

The version data displayed under “NCU Version” state the software versions on the CF card. This especially applies to measurement cycles and cycle versions.

- Version data, NCU
- Version data, HMI files
- Version data, cycles
- Version data, definitions

The measurement cycles and cycles that are used by HMI-Advanced are available on the hard disk of the PCU. Their version can be queried via the horizontal softkeys.

With the “Detail” softkey, you can query further information about a version, for example, to pass it on to hotline support personnel if service is required.



```
Version data NCU/NCK
NCK
 Version                                67.02.01
 InternalVersion                        67.02.01

 Numeric Control                        67.02.01
 NCK-SI                                 67.02.00
 NCKS710-6a2c                           67.02.01
 NCKS710-6a2c                           67.02.01
 NCKS710-6a2c                           67.02.01
 NCKS710-6a2c                           67.02.01
 NCKS710-6a2c                           67.02.01
 License                                67.02.01
 Joblist for IBN                        02.04.00
 Joblist for IBN/KOMP                   02.04.00
 Joblist for UPGRADE                    02.04.00
 Base System                            67.01.00
 nrk710.abb                             01.09.07
  RtExt-process                         05.03.00.00
  RtExt-module                          05.03.00.00
  RtExt-module                          05.03.00.00
```

Figure 5-3 NCU version (example)

5.4 Setting-up and upgrading the system

5.4.1 Series commissioning

Series commissioning

The "series commissioning" function is used to back-up (archive) the data of a control. This data back-up can also be transferred to other controls so that they are all brought into the same state. In this case, NC, PLC, drive and - if required, HMI data - can be separately saved or together in back-up (archive) files and then downloaded again from there. For NC data, optional compensation data of the axes can also be saved.

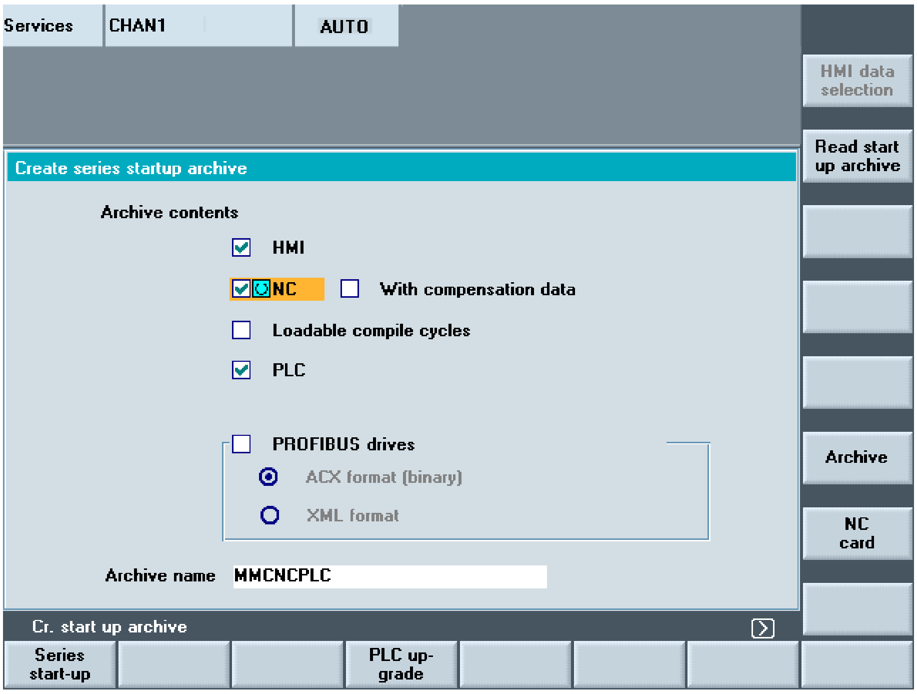
The drive data cannot be changed and can be optionally saved in the acx format as binary data or in the xml format.

Requirements

Access authorization is required to carry-out series commissioning. The access authorizations and other settings for the archive are assigned in the file DINO.INI.

The following access levels are preset:

- From access level 3 onwards (password): To generate an archive for series commissioning.
- From access level 4 (key-operated switch): To download an archive for series commissioning.



Create series startup archive

Archive contents

☑ HMI

☑ NC □ With compensation data

□ Loadable compile cycles

☑ PLC

□ PROFIBUS drives

● ACX format (binary)

○ XML format

Archive name MMCNCPLC

Figure 5-4 Creating a series commissioning archive

Creating an archive

To create a commissioning archive:

1. In the operator area "Services" select "Series commissioning" on the expanded softkey bar.
2. Select archive content.

   If HMI data are selected for the archive:

   The F:USER directory is always backed up.

   – Keep the default setting with the standard values.

   – Select the required data using the softkey "HMI data selection". Directories \ADD_ON and \OEM can be saved under "Additional products", either completely ("Complete") or just the INI files ("Configuration").
3. Archive name: Accept the recommendation given by the system or enter a name.
4. Start creating the archive using the softkey "Archive" or "NC card".

   Ensure that there is sufficient free memory space.

   Result: When selecting "Archive", the archive is saved on the hard disk under the following path: `F:\dh\arc.dir\ ...`

   Alternatively, the archive can be saved on a partitioned USB Flash Drive (with a partition for Linux and Windows).

Read-in archive

Requirements: The commissioning archive must be located under the same path on the new control as on the control on which it was created.

In order to read-in a commissioning archive:

1. Select the softkey "Read-in commissioning archive": The directory is displayed.
2. Start the read-in (download) operation using the softkey "Start".
3. Acknowledge the prompt with "Yes".

   The archive is read-in (downloaded) and the system is automatically re-started in order that the archive data become effective.

Note

When importing an archive, the available space on the hard disk must be at least the size of the series commissioning archive.

If a control is configured for M:N, the PCU must be re-started after executing a series commissioning (start-up) so that the devices in the network (NCU, PLC, HMI) can re-synchronize.

For compile cycles (CC) that can be downloaded, the following specifically applies:

The compile cycles that can be loaded are archived in the series commissioning under the following requirements:

- In the commissioning archive, in the target directory ../NC_CARD.DIR/CCOEM.DIR, there must be at least one compile cycle.
- It is only possible to selectively archive individual compile cycles by copying to a USB memory.

5.4.2 Upgrading the PLC

Requirements

Access level 0-3 is required both to upgrade the PLC as well as to upgrade system data blocks (SDB).

Use

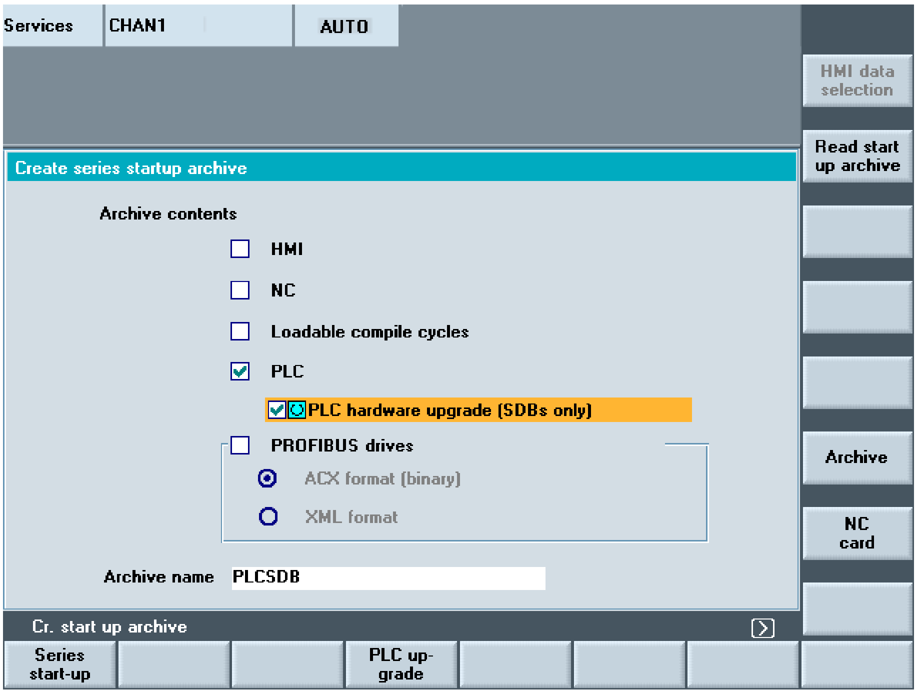
When upgrading the PLC and/or system data blocks, only specific data can be updated by entering the blocks into an INI file that are not overwritten and should therefore be updated:

- **Upgrading the PLC**

  In order to upgrade the PLC, select "PLC" and create a PLC series commissioning archive. Further, enter the blocks into a separate INI fie that should not be updated.
- **Upgrading system data blocks**

  To upgrade the hardware, under "PLC", also select "PLC hardware upgrade (only SDB)". This means that the hardware configuration can be made without using STEP 7 and using the saved SDB upgrade archive.

  This option can only be selected if exclusively the "PLC" option is selected. The recommended archive name is SDB.ARC. All SDB of the actual PLC, which are in the RAM and active, are entered into this archive. SDB that are set-up by the CPU as default setting are not entered.



Create series startup archive

Archive contents

☐ HMI

☐ NC

☐ Loadable compile cycles

☑ PLC

☑ PLC hardware upgrade (SDBs only)

☐ PROFIBUS drives

● ACX format (binary)

○ XML format

Archive name PLCSDB

Figure 5-5 Selecting system data blocks

Structure of the INI file

The data blocks are entered into a descriptive file with the *.INI file extension; their contents should be kept when upgrading.

The contents of this file are structured line-by-line as follows:

`;message=<Text>`	;	Language-independent text for the dialog line (optional)
`DB<number>`	;	The contents of the complete DB <number> are kept (1)
`DB<from> - <to>`	;	The contents of all DBs from <from> to <to> are kept (2)
`DB<number> [<byte>]`	;	The byte of the data block <number> is kept (3)
`DB<number> [<from> - <to>]`	;	The bytes <from> to <to> of the data block <number> are kept (4)

- User-specific comments can be specified. For an error-free descriptive file, this language-independent text is output in the dialog line and should not be longer than 50 characters so that it can be completely displayed in this dialog line. Only the first text instruction is evaluated.
- Empty lines and comments are possible - starting with a semicolon. Separators (blanks or tabs) can be located but do not have to be located between the symbols for the data blocks (DB), the addresses, the end of range characters (-) and the byte ID [ ]. A differentiation is not made between upper/lower case letters.

- If ranges of data blocks are addressed (2), then it is not permissible to specify additional byte addresses or byte range, otherwise an error message is output. If byte addresses are defined for a data block (3, 4), then for this data block there may be no entry without byte address (1, 2) otherwise an error is output.
- The instructions can be in any sequence.

Syntax check of the INI file

The descriptive file with the *.INI file extension can be copied into the archive directory via a USB drive or network drive (option). When displaying the files under PLC upgrade, the syntax of the data contents is checked. If a syntax error is identified, the line number of the first syntax error is displayed in the dialog line. When opening the file, the cursor is located at the line with the error.

Upgrade sequence

In order to upgrade the PLC, proceed as follows:

- Before upgrading a complete PLC, create a series commissioning archive of the machine to be upgraded; if an error occurs, the original state of the PLC will be able to be restored.
- For the PLC upgrade, a complete PLC series commissioning archive is created as well as a descriptive file, e.g. generated by the machine construction OEM. The archive and the descriptive file are downloaded into the archive directory of the machine to be upgraded. The target directory is mandatory.
- Select the INI file and a PLC series commissioning archive from which the archive directory is selected. A syntax check is subsequently started.
- Using the softkey "Start", data is read out of the archive into the PLC using the descriptive file. To do this, initially, a complete, temporary original PLC archive of the actual machine is generated. A complete temporary target PLC archive is generated from this archive using the selected upgrade archive and the descriptive file.The length of the data blocks is determined by the upgrade archive and the contents are determined by the descriptive file. The temporary target PLC archive is then downloaded and the two temporary archives are again deleted after the upgrade.
- If several machines with identical PLC status are to be upgraded, then from the first upgraded machine, a complete PLC archive can be generated and using this PLC archive, the remaining machines can be commissioned without requiring a descriptive file.

  Before upgrading using such an archive, the following prompt is output:

  ```
  Series commissioning archive: Upgrade hardware configuration PLC
  (SDBs) ?
  ```

Note

The softkey "Start" is only effective if an archive and a descriptive file with the correct syntax are selected.

5.4.3 Example: How to upgrade a PLC

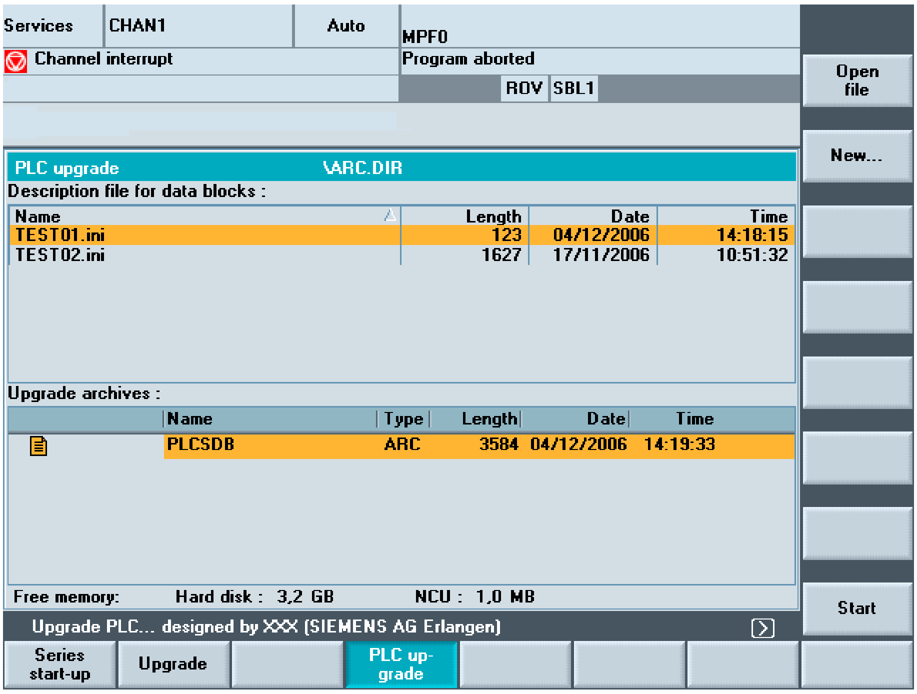
Procedure

In the following example, data blocks DB3, DB67 and DB111 are not to be overwritten.

1. As back-up, create a series commissioning archive of the PLC data.
2. Create a descriptive file with the following contents:

```
;Message= designed by XXX (SIEMENS AG Erlangen)
DB3[3-78]
DB67 ; DB67 complete
DB 3 [79]
DB3[2]

DB111 ; DB111 complete
```

3. Save the descriptive file as TEST1.INI and save in the directory ..\dh\arc.dir of the control that is to be upgraded.
4. Select the descriptive file TEST1.INI and the series commissioning archive to upgrade the control. The syntax is checked.
5. Press "Start": While copying, the text message is output from the INI file.



Channel interrupt
Program aborted

PLC upgrade \ARC.DIR

Description file for data blocks :

Name	Length	Date	Time
TEST01.ini	123	04/12/2006	14:18:15
TEST02.ini	1627	17/11/2006	10:51:32

Upgrade archives :

Name	Type	Length	Date	Time
PLCSDB	ARC	3584	04/12/2006	14:19:33

Free memory: Hard disk : 3,2 GB NCU : 1,0 MB

Upgrade PLC... designed by XXX (SIEMENS AG Erlangen)

Figure 5-6 Start PLC upgrade

Result

All of the data blocks of the upgrade archive are transferred into the PLC for which no entry exists in the descriptive file: The PLC was upgraded to the current status without overwriting data blocks DB3, DB67 and DB111.

Rules

For the data blocks listed in the descriptive file, the following behavior is defined when upgrading:

- If the data block of the upgrade archive does not exist in the PLC, then an appropriate entry is made in the error log.
- The length of the data block is defined by the length in the upgrade archive.
- If byte addresses are defined for the data block, then these data areas are transferred from the upgrade archive into the PLC. Error messages are output if the address either does not exist in the upgrade archive or in the PLC.
- If no byte addresses are defined for the data block, then the contents of the PLC data block are kept. Where necessary, the data block is shortened to the new length.
- If the data block in the upgrade archive is longer than that in the PLC, then the remaining range is filled with the contents of the upgrade archive.

5.5 Service displays

Overview of service displays

When "Service Displays" is selected the "Service Overview" window is displayed. Here, the various operating states are indicated for each machine axis by traffic lights.

A selection of more displays for diagnosing faults also appears:

- Service axis
- System resources
- Configuration Data
- Communications error log
- Action log

Service axis

The information in the "Service axis" display is used to check the following values:

- checking the setpoint branch (e. g., position setpoint, speed setpoint, spindle speed setpoint prog.)
- checking the actual value branch (e. g. actual position value measuring system 1 and 2, actual speed value)
- optimizing the position control loop of the axis (e. g. following error, control deviation, servo gain factor)
- checking the entire control loop of the axis (e. g. through position setpoint/actual-value comparison and speed setpoint/actual-value comparison)
- checking hardware errors (e.g. by checking the encoder: If the axis is moved mechanically, the actual position value must change.)
- Checking and setting the axis monitoring functions.

References: Description of Functions, Basic Functions; Diagnostic Tools (D1)

System resource display

The following system resources for the NCU are displayed in the window "NC utilization":

- The net and overall runtime of the position controller, the interpolator and the pre-run in milli-seconds
- NCU load level as a percentage
- Buffer filling level as a percentage

The display update is stopped using the "Stop" softkey, the displayed values are updated again with the "Start" softkey.

Output of configuration data

The configuration data of a machine (HMI version, NCU version, axis configuration, drive configuration, bus parameters, active bus nodes) can be written into a file and subsequently read out or printed.

Configuration data is output in 2 stages:

1. Creation of the configuration data file CONFIGURATION_DATA in the "Diagnostics" operating area by pressing the "Config. data" softkey.
2. Read-out of the CONFIGURATION_DATA file in the “Services” operating area. To do this, the configuration file CONFIGURATION_DATA is generated.

The configuration data are combined in the CFGDAT.TXT file. The path and name of the file are output in the info line.

5.5.1 Service axis

Use

The information in the "Service axis" dialog is used to check:

- Setpoint branch: e.g. position setpoint, speed setpoint, programmed spindle speed setpoint
- Actual value branch: e.g. position actual value measuring system 1 and 2, position setpoint
- Optimizing the position control loop of the axis: e.g. following error, system deviation, Kv factor)
- Complete control loop of the axis: e.g. by comparing the position setpoint and the position actual value, speed setpoint and speed actual value
- Hardware faults: e.g. checking the encoder. If the axis is moved mechanically, the actual position value must change.
- Setting and checking axis monitoring functions

The "Service axis/spindle" dialog displays reference values and units for the machine axis together with axis name and axis number.

The service values of the next or previous axis are displayed using the softkeys "Axis +" and "Axis −". The "Direct selection" softkey allows you to select an axis directly from the list of available axes.

5.5.2 Displaying system resources

Function

The system resources for the NCU are displayed in the dialog "NC utilization":

- The net and overall runtime of the position controller, the interpolator and the pre-run in milli-seconds
- NCU load level as a percentage
- Buffer filling level as a percentage

The display update is stopped using the "Stop" softkey, the displayed values are updated again with the "Start" softkey.

5.5.3 Output of configuration data

Function

The configuration data of a machine (HMI version, NCU version, axis configuration, drive configuration, bus parameters, active bus nodes) can be written into a file and subsequently read out or printed.

Configuration data is output in 2 stages:

- Creation of the configuration data file CONFIGURATION_DATA in the "Diagnostics" operating area by pressing the "Config. data" softkey.
- Read-out of the CONFIGURATION_DATA file in the “Services” operating area. To do this, the configuration file CONFIGURATION_DATA is generated.

Procedure:

1. Select the menu "Service displays".
2. The horizontal softkey bar changes
3. Press the "Config. data" softkey.

   The system gathers the configuration data, writes it into the file CFGDAT.TXT and displays it. The path and name of the file are output in the info line.

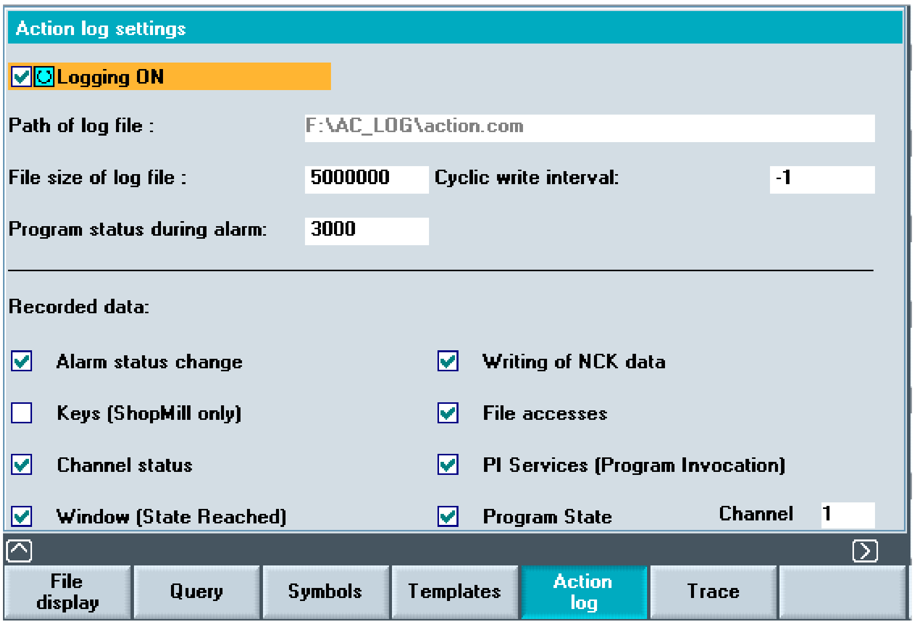
5.6 Action log

5.6.1 Setting the action log

Overview

The "action log" function can be used to log operating sequences. By default, if the action log is enabled, the data that can be logged are recorded. The action log setting is password-protected (access level 3: end user).

Using the operating area "Start-up" → "HMI" → "System Settings," select which actions of the control should be logged in the "Action log settings":


Action log settings

☑ Logging ON

Path of log file : F:\AC_LOG\action.com

File size of log file : 5000000 Cyclic write interval: -1

Program status during alarm: 3000

Recorded data:

☑ Alarm status change ☑ Writing of NCK data

☐ Keys (ShopMill only) ☑ File accesses

☑ Channel status ☑ PI Services (Program Invocation)

☑ Window (State Reached) ☑ Program State Channel 1

Figure 5-7 Action log settings

Logging on/off

Additional settings:

- The path of the log file is permanently set and cannot be changed.

• File size of log file:	in bytes
• Transmission Write interval:	− 1: The log is only saved to hard disk upon request, e.g. trigger, (default setting). > 0: The longest waiting time in seconds before a log file is saved on hard disk.
• Program status when an alarm occurs:	Alarm number entry separated by a comma: The alarms entered here are immediate triggers for immediately saving the current action log on the hard disk.

Logged data

The following data are logged:

- Alarm status change: Alarms and PLC messages, incoming and outgoing. All of the displayed alarms and messages can be logged.
- Keys (only ShopMill): For HMI-Advanced, this selection has no function.
- Channel status: Changing the channel status and the setting of the override switch: NC/PLC states are logged. If this information can be obtained in a timely manner, in many cases these states can be used to verify the operation of the MCP.
- Window (state reached): Using the softkeys and function keys to select screens and to change the operator area
- Writing NCK data: All variables that are written via the input field (e.g. write access operations to geometrical data such as tool offsets and work offsets can be logged. A description of the variables can be obtained via the Help function in the Parameters operating area under Variable Views.
- File accesses: All domain services (e.g. load/unload program) are logged.
- PI services: (Program Invocation) All PI services (e.g. program selection, file deletion, tool deletion) are logged. A description of the variables can be obtained via the Help function in the Parameters operating area under Variable Views.

With NCK version 7.4 and higher and NCU system software version 1.4 and higher, the following applies to solution line:

- Program state / channel: The following data are logged for the events NC Start, NC Stop, Reset, PLC Signal (trigger for the file CRASH.COM):

  NC Start, NC Stop and Reset Program:

  - The current main program that is selected (only AUTO)
  - Complete information about the program levels (all assigned program levels with the program names including directory)
  - Actual NC block pointer (only AUTO and MDI) at all levels

PLC signal:
(initiates writing to file CRASH.COM)

- Operating mode
- Actual main program that is selected (only AUTO)
- Actual block that is active (only AUTO and MDI) and its predecessor and successor (three block display).
- Complete program level information (all assigned program levels with the program names including directory)
- Actual NC block pointer (only AUTO and MDI) all levels
- All alarms and messages that are present
- Actual position of all of the axes in the WCS
- Actual position of all axes in the MCS
- Distance to go of all axes
- Program velocity (only AUTO and MDA)
- Programmed spindle speed (only AUTO and MDA)
- Actual spindle speeds
- Override influence of the axes and spindles
- Actual tool and active tool offset
- All active G functions
- All active frames, TRAFOS... if not clearly visible from the G function.
- All active M functions

Machine data for the action log

The machine data should be set to the following values:

General MD:		Value	Significance
11295	PROTOC_FILE_MEM[9]	 0 1	Memory type of the log files: = SRAM = DRAM for solution line
18370	MM_PROTOC_NUM_FILES[9]	3	Maximum number of log files
18371	MM_PROTOC_NUM_ETPD_STD_LIST[9]	3	Number of standard ETPD data lists

Channel MD		Value	Significance
28300	MM_PROTOC_USER_ACTIVE[9]	1	Activating logging for a user
28302	MM_PROTOC_NUM_ETP_STD_TYP[9]	3	Number of ETP standard event types

Configuration file ACTLOG.INI

All settings of the action log are stored in file user\actlog.ini. Display MD 9012 ACTION_LOG_MODE has no meaning.

See also

ACTLOG.INI (Page 274)

5.6.2 Structure of the log file

Overview

The file ACTION.COM is generated if the action log is activated, i.e. data logging is enabled. The file ACTION.COM is a binary file and is always repeatedly overwritten as ring-type buffer. The most recent results are located at the beginning of the file.

An empty binary file CRASH.COM is present. An intermediate status of the log ACTION.COM is saved in the file CRASH.COM. Writing is initiated by setting the interface signal DB19.DBX0.6 or write trigger using an alarm. This contents are kept up to the next trigger.

Content of the log file

The log file comprises the following columns:

- Date
- Time
- Name
- Event ID
- Event description

Example of a log file:

```
 HMI version:  06.04.24.00              NCK version:  660000
Activated entries: HMI_START HMI_EXIT PLC_CRASH PLC_CRASH_ALARM
KEY_PRESSED KEY_PRESSED KEY_PRESSED KEY_RELEASED KEY_RELEASED
KEY_RELEASED ALARM ALARM_QUIT OPEN_WINDOW OPEN_WINDOW
OPEN_WINDOW CLOSE_WINDOW CLOSE_WINDOW CLOSE_WINDOW
CH_STATE_CHANGED OPMODE_CHANGED TOOL_CHANGED OVERRIDE PI_CMD
DOM_CMD WRITE_VAR WRITE_VAR WRITE_VAR FINDBL_CMD OVERSTORE
FILE_ACCESS AREA_CHANGED USER ACTIVATED DEACTIVATED SUSPEND
RESUME

 --- Date         Time        User              Entry Id
 Entry
===============================================================
--- 30.03.2006 13:26:04 HMI-Adv. 0 WRITE_VAR
NCK/PLC Var. wrote: /plc/datablock/byte[c19,0] = 64
--- 30.03.2006 13:26:04 <default> 0 OPEN_WINDOW
Windowstate changed: State 9, Index 15 in Application "DG"
```

```
--- 30.03.2006 13:26:04 HMI-Adv. RE 0 KEY_PRESSED
Key pressed: VSK 7 SHIFT+F8 ( 14)
--- 30.03.2006 13:26:01 <default> 0 OPEN_WINDOW
Windowstate changed: State 9, Index 13 in Application "DG"
--- 30.03.2006 13:26:01 HMI-Adv. RE 0 KEY_PRESSED
Key pressed: VSK 5 SHIFT+F6 ( 12)
--- 30.03.2006 13:25:59 <default> 0 OPEN_WINDOW
Windowstate changed: State -990, Index -1 in Application "RE"
--- 30.03.2006 13:25:59 <default> 0 OPEN_WINDOW
Windowstate changed: State -992, Index -1 in Application "RE"
--- 30.03.2006 13:25:59 HMI-Adv. RE 0 KEY_PRESSED
Key pressed: HSK 4 F5 ( 5)
--- 30.03.2006 13:25:04 <default> 0 OPEN_WINDOW
Windowstate changed: State 1, Index -1 in Application "DG"
--- 30.03.2006 13:25:03 <default> 0 OPEN_WINDOW
Windowstate changed: State -992, Index -1 in Application "RE"
--- 30.03.2006 13:25:03 HMI-Adv. RE 0 KEY_PRESSED
Key pressed: HSK 4 F5 ( 5)
--- 30.03.2006 13:25:02 HMI-Adv. RE 0 KEY_PRESSED
Key pressed: AREA-SWITCH ( a)
--- 30.03.2006 13:24:56 HMI-Adv. 0 CH_STATE_CHANGED
Channel State changed to active
--- 30.03.2006 13:24:56 HMI-Adv. 0 TOOL_CHANGED
Active tool changed to 0, Duplo 1

Mode: AUTO Program: stopped Channel: interrupted
Program-Level information:
Level Program Invoc Offset
running:
1 /_N_MPF_DIR/_N_MAINPROG_MPF 1 0
2 D:/TEST/MYSUB.SPF 1 0
stopped:
1 /_N_MPF_DIR/_N_MAINPROG_MPF 0 -1
2 D:/TEST/MYSUB.SPF 0 0

Actual Block:
T0X0Z0
Number of Machine axis: 6
MCS-Name Position DistToGo
X1 46.300 0.000
Y1 108.600 0.000
Z1 4439.366 0.000
A1 319.864 0.000
B1 114.935 0.000
C1 0.000 0.000
```

```
Zero-Offset Shift Fine Mirror Rotation Factor
ACTFRAME 24.000000 22.300000 0 0.000000 1.000000
IFRAME 56.000000 3.300000 0 0.000000 1.000000
PFRAME 4398.000000 34.340000 0 0.000000 1.000000
EXTFRAME 0.000000 0.000000 0 0.000000 1.000000
TOTFRAME 0.000000 0.000000 0 0.000000 1.000000
ACTBFRAME 0.000000 0.000000 0 0.000000 1.000000
SETFRAME 12.000000 0.300000 0 0.000000 1.000000
EXTSFRAME 33.000000 2.300000 0 0.000000 1.000000
PARTFRAME 4354.000000 34.000000 0 0.000000 1.000000
TOOLFRAME 0.000000 0.000000 0 0.000000 1.000000
MEASFRAME 0.000000 0.000000 0 0.000000 1.000000
WPFRAME 0.000000 0.000000 0 0.000000 1.000000
```

Additional information on "State" and "Index" is provided in:

References: HMI Programming Package, Part 1 Operator's Guide → "Sequence control"

5.6.3 How to save the log file

Requirements

The actual log file is displayed in the operator area "Diagnostics" → "Service Displays" → "Action Log" on the screen (softkeys "MMCWIN file" and "Crash Log").

Log file after initiating the trigger

The file CRASH.COM is generated in the following cases:

- If the interface signal DB19.DBX0.6 "Save action log" changes from 0 to 1.
- If the alarm entered in field "Write trigger via alarms" occurs.

The PLC must set and reset the bit at the user interface (in the same way as the key lock and screensaver control, for example). HMI-Advanced evaluates the rising edge of the signal (i.e. a transition from 0 to 1) and creates the log file when the signal arrives. HMI-Advanced only read-accesses the data, i.e. the user must reset the signal in the PLC program.

If the bit is already set when the HMI starts up, HMI-Advanced does not respond to the value of the bit (until it is reset and set again).

Saving logs

To read out the log files, proceed as follows:

1. In order to save the log files, select "Save log".

   The following files can be saved:

   – ACTION.COM (binary file)

   – ACTION.LOG

   – CRASH.COM (binary file)

   – CRASH.LOG

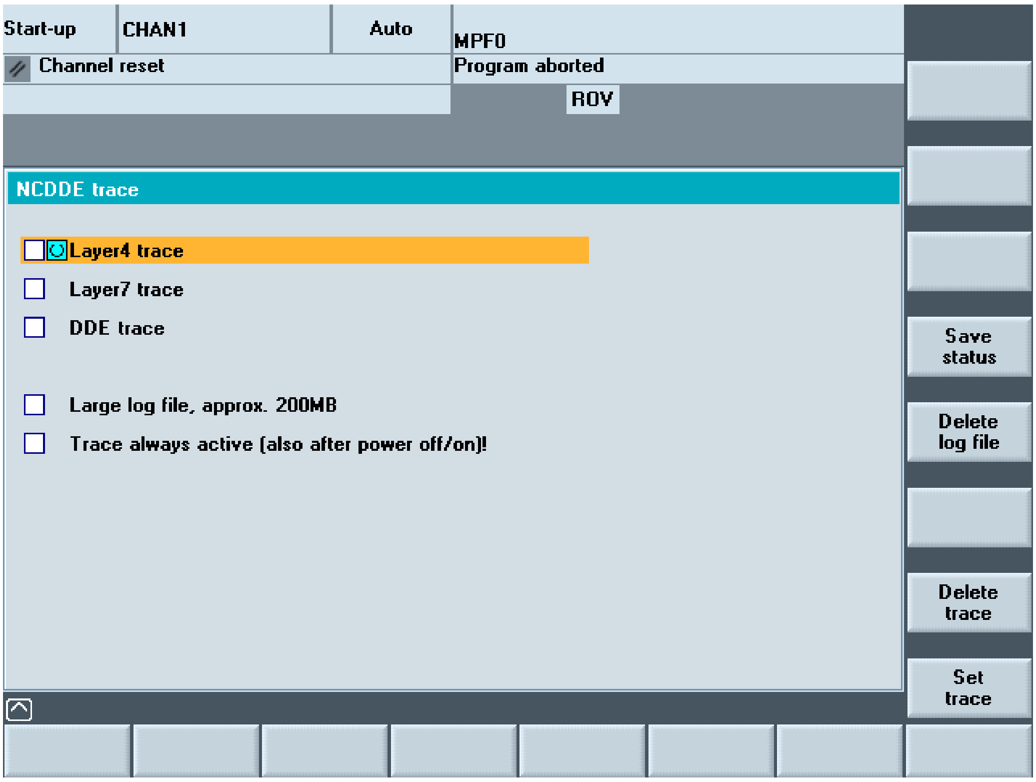
2. The logs are saved under ..\Diagnosis\Log files.

For example, select "Services" → "Data Out" → "Disk", in order to read out the log file.

5.6.4 "NCDDE Trace" log

Diagnostics with "NCDDE Trace "

Press the "Trace" softkey to open the following dialog:



NCDDE trace

□ Layer4 trace

□ Layer7 trace

□ DDE trace

□ Large log file, approx. 200MB

□ Trace always active (also after power off/on)!

Figure 5-8 "NCDEE Trace" dialog

The Trace options should be selected in consultation with system or OEM development personnel.

The log created is used to investigate software problems and must be sent to system or OEM development personnel for analysis and diagnostics purposes.

Creating the log file

The log file is called `debug.out` or `_debug.out` and is saved in the following directory: `..\mmc2`

5.7 HMI Analyzer

5.7.1 Using the HMI Analyzer

Use

The HMI Analyzer diagnoses any problems occurring on a PCU 50.3. The result of the analysis is displayed via the HMI Analyzer in lists. It can also be stored in ASCII files and analyzed later. In this way, the differences between the supplied versions and a customer-modified version can be determined.

Function

All settings of the INI files of HMI-Advanced are analyzed and the resulting settings are displayed.

Further analyses are possible via the listing of DDE messages, monitoring of the processes, memory utilization and network settings.

Configuring the HMI Analyzer

The HMI Analyzer is configured in the HMIAnalyzer.ini file:

```
[HMI_DIRS]
DIR01 = MMC2b
DIR02 = HMI_ADV
DIR03 = ADD_ON
DIR04 = OEM
DIR05 = USER

[HMI_FILES]
FILE01 = MMC.INI
FILE02 = REGIE.INI
FILE03 = MBDDE.INI
FILE04 = OEMFRAME.INI
FILE05 = NETNAMES.INI
FILE06 = KEYS.INI

[Additional]
FILE01 = F:\Programs\Test1.ini
FILE02 = F:\Programs\Test2.ini
```

Section [HMI_DIRS]

All the directories that are to be used for analyzing the INI files are entered here.

Section [HMI_FILES]

All the INI files that are to be used for analysis are entered here. At the same time, the files in the directories that are entered under HMI_DIRS are searched and analyzed.

Section [Additional]

Additional INI files that are not subject to the INI concept of HMI-Advanced, i.e. INI files from add-on applications, can be entered here. The complete path is needed for the entries. Only the file of this path is analyzed.

5.7.2 Using the HMI Analyzer

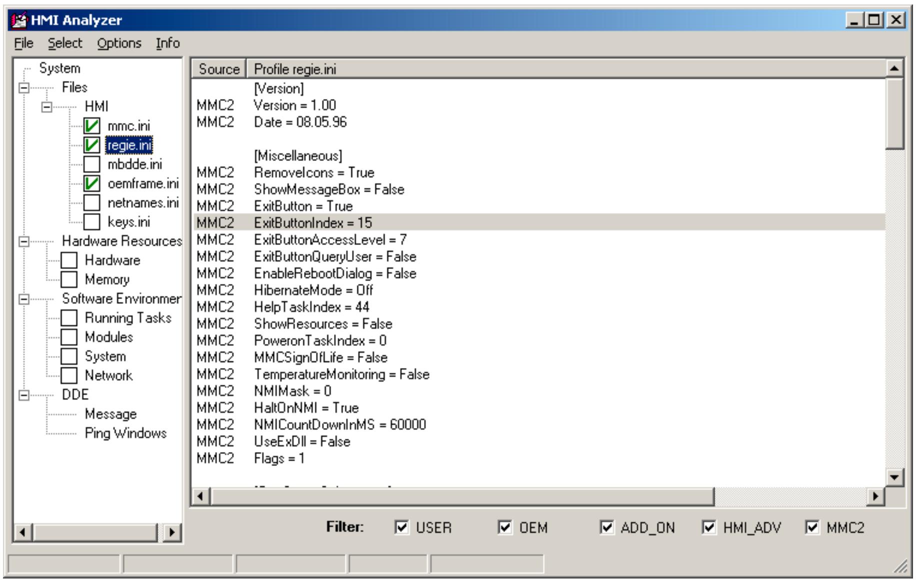
Overview

Use the following link on the Windows desktop to start the HMI Analyzer:

HMIAnalyzer

When the HMI Analyzer is started, the settings in the initialization file HMIAnalyzer.ini are analyzed. The point selected in the tree to the left is analyzed. The result of the analysis is displayed as a list on the right side.

The result of the analysis can be saved as a "profile". All of the items marked in the tree are saved (toggle key or double-click on the respective box).



```
Source  Profile regie.ini
        [Version]
MMC2    Version = 1.00
MMC2    Date = 08.05.96

        [Miscellaneous]
MMC2    RemoveIcons = True
MMC2    ShowMessageBox = False
MMC2    ExitButton = True
MMC2    ExitButtonIndex = 15
MMC2    ExitButtonAccessLevel = 7
MMC2    ExitButtonQueryUser = False
MMC2    EnableRebootDialog = False
MMC2    HibernateMode = Off
MMC2    HelpTaskIndex = 44
MMC2    ShowResources = False
MMC2    PoweronTaskIndex = 0
MMC2    MMCSignOfLife = False
MMC2    TemperatureMonitoring = False
MMC2    NMIMask = 0
MMC2    HaltOnNMI = True
MMC2    NMICountDownInMS = 60000
MMC2    UseExDll = False
MMC2    Flags = 1
```

Filter: ☑ USER ☑ OEM ☑ ADD_ON ☑ HMI_ADV ☑ MMC2

Figure 5-9 Example of .ini files

Saving the current data (profile)

The profile of the currently selected items can be saved via the menu item "File" → "Save Profile". The menu item is only active if items for saving in the profile are marked in the tree.

When saving, a sub-directory is created in the directory in which the HMI Analyzer is located. The name of the directory is composed of the date and time at the time the data was saved.

Example:

"20060122_083755" for saving on January 22, 2006 at 08:37:55. All of the INI files marked at this time in this directory are saved as profiles.

The INI files saved as profiles are the "resultant" INI files. In this way, the profile shows the current valid configuration.

The result files can be considered individually in an external ASCII editor.

Deleting the current data (profile)

Individual profiles can be deleted from the hard disk via the menu item "File" → "Delete Profile".

Comparison of two saved profiles

By selecting the menu item "File" → "Compare" in the main window, two saved profiles (configurations) can be compared to each other.

All previously saved profiles are listed in the "Available Profiles" list. The profile selected in the list is selected for comparison using the the key "Add **à**" and moved to the "Compared Profiles" list. The first selected profile is considered as the original, the second selected profile is considered the modified profile. A profile can be moved back into the list using the "**ß** Remove" button. The sequence in the list can be changed via arrow right.

The "Compare" button can be used as soon as two profiles in the "Compared Profiles" list are selected for comparison. After the comparison is done, the result is displayed in the main window of the application and the selection window is closed.

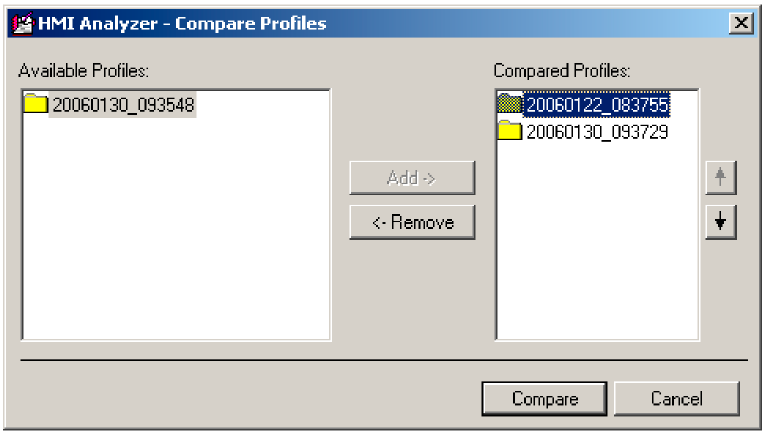


Figure 5-10 Comparing profiles

The result of the comparison can be seen in the main window of the application. The result of the comparison can also be saved in an ASCII file via the menu items "File" → "Save Comparison".

A saved comparison result can be displayed again by the HMI Analyzer via the menu items "File" → "View Comparison".

5.7.3 Analyzing INI Files

System → Files

All entries of the analyzed INI files are displayed here.

HMI Display of the corresponding INI entries from files that are subject to the INI concept of HMI-Advanced. The directory from which the entry comes is also displayed ("Source" column). For this, all the directories that are entered under HMI_DIRS in HMIAnalyzer.ini are scanned.

Additional files Display of the addition INI files that are not subject to the INI concept, i.e. these INI files are only searched for in the specified path.

System → Hardware Resources

Hardware Under hardware, parts of the existing hardware are listed, e.g. information on the processor and information on the individual hard disk partitions and drives.

Memory More precise information on the existing and available memory is displayed here in KB:

Used Memory:	Percent of allocated memory
Physical Memory, total:	Total size of the RAM
Physical Memory, available:	Free, available RAM memory
Page File, total :	Total memory of the swap file
Page File, available:	Free, available memory of the swap file
Virtual Memory, total:	Total size of the virtual memory
Virtual Memory, available:	Free, available virtual memory

System → Software Environment

Running Tasks The memory usage and the running of all active processes are displayed here.

The following columns are displayed:

Name	Name of the process
Running	"X" process is running '-' process was ended
ProcessID	Process ID
WorkingSetSize	Current memory usage in kB
DiffWorkingSetSize	Modification of the memory since the start of the HMI Analyzer or since the last reset

PagePoolUsage	Swap memory in KB
DiffPagePoolUsage	Modification of the swap memory since the start of the HMI Analyzer or since the last reset
PageFaults	Access violations of the swap file
DiffPageFaults	Modification of the access violations of the swap file since the start of the HMI Analyzer or since the last reset

Additional settings (below):

Cyclic	The view is cyclically updated. The differences since the start or since the last reset are computed and displayed. The update rate can be specified in milliseconds [ms] to the left next to the "Cyclic" selection.
Reset	Resets the differential values. This action can only be carried out for cyclic display.
Single	The view is updated once as a static display. The memory requirement of all processes is re-evaluated.

Modules When the Modules item is selected, the loaded DLLs for all running processes are determined. When HMI-Advanced is running, the applications of all operating areas that are entered with ProLoad;=False in the Regie.ini are also started.

For this, the loaded DLLs are also determined.

System Gives information on general PCU or PC data

Network Display of the current network data:

All the data that can be determined via the command "ipconfig /all" is displayed. Für das Betriebssystem Windows NT wird zusätzlich noch ermittelt, ob NetBeui installiert ist.

Note

Operating areas that are entered with PreLoad:=False in REGIE.INI and cannot be found as a process are started by the HMI Analyzer before the analysis. The start is done via the master control server (Regie server). During this, all the operating areas to be started are activated by "SwitchToTask". After all the operating areas are started, the operating area changes to "Machine".

When setting up a hotlink via the DCTL, a data query request is also sent via the DCTL. The HMI Analyzer cannot display the current value for this request ("Value" column) .

5.7.4 Application Examples

(1) OEM application that is linked-in does not start

Problem:

An OEM application was linked into HMI-Advanced, but it does not start after actuating the corresponding softkey or it starts another application.

Possible cause:

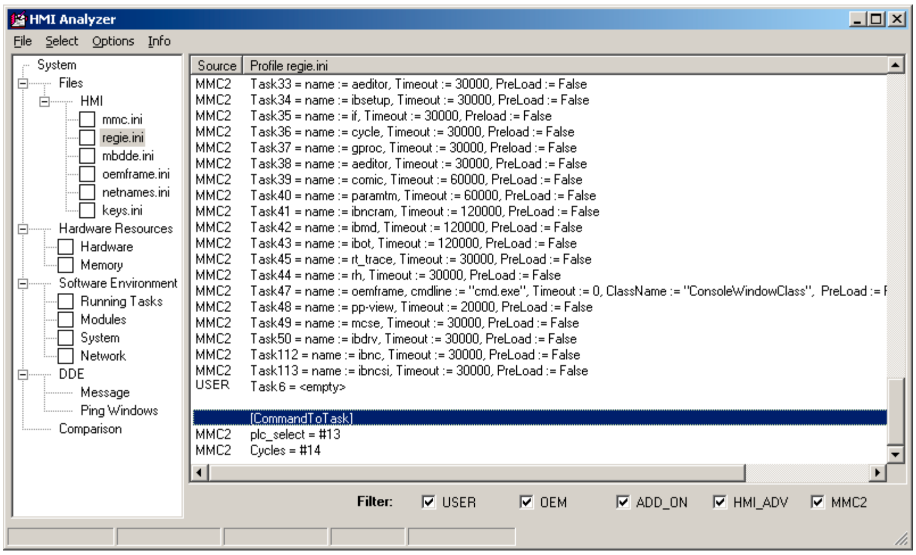
Due to the hierarchical execution of the HMI directory, the entry made in the Regie.ini file was overwritten by an entry in another regie.ini file with a higher priority and is therefore not in effect.

For example, an entry was made in the OEM directory:

```
[TaskConfiguration]
Task6 = name := TestApp, Timeout := 10000, PreLoad := True
```

This is then overwritten in the user directory:

```
[TaskConfiguration]
Task6 = <empty>
```



```
Source  Profile regie.ini
MMC2    Task33 = name := aeditor, Timeout := 30000, PreLoad := False
MMC2    Task34 = name := ibsetup, Timeout := 30000, PreLoad := False
MMC2    Task35 = name := if, Timeout := 30000, Preload := False
MMC2    Task36 = name := cycle, Timeout := 30000, Preload := False
MMC2    Task37 = name := gproc, Timeout := 30000, Preload := False
MMC2    Task38 = name := aeditor, Timeout := 30000, PreLoad := False
MMC2    Task39 = name := comic, Timeout := 60000, PreLoad := False
MMC2    Task40 = name := paramtm, Timeout := 60000, PreLoad := False
MMC2    Task41 = name := ibncram, Timeout := 120000, PreLoad := False
MMC2    Task42 = name := ibmd, Timeout := 120000, PreLoad := False
MMC2    Task43 = name := ibot, Timeout := 120000, PreLoad := False
MMC2    Task45 = name := rt_trace, Timeout := 30000, PreLoad := False
MMC2    Task44 = name := rh, Timeout := 30000, PreLoad := False
MMC2    Task47 = name := oemframe, cmdline := "cmd.exe", Timeout := 0, ClassName := "ConsoleWindowClass",  PreLoad := F
MMC2    Task48 = name := pp-view, Timeout := 20000, PreLoad := False
MMC2    Task49 = name := mcse, Timeout := 30000, PreLoad := False
MMC2    Task50 = name := ibdrv, Timeout := 30000, PreLoad := False
MMC2    Task112 = name := ibnc, Timeout := 30000, PreLoad := False
MMC2    Task113 = name := ibncsi, Timeout := 30000, PreLoad := False
USER    Task6 = <empty>

        [CommandToTask]
MMC2    plc_select = #13
MMC2    Cycles = #14
```

Filter: ☑ USER ☑ OEM ☑ ADD_ON ☑ HMI_ADV ☑ MMC2

Figure 5-11 Checking the start of the OEM application

HMI Analyzer:

Under Files → HMI, the regie.ini is selected. The resulting regie.ini file from all the HMI directories is now displayed (mmc2, hmi_adv, add_on, oem, user). Now it becomes clear whether the entry that was made is also truly active or was overwritten by another entry. In addition, it is possible to recognize which HMI directory the entry comes from.

(2) No connection to the NCU/PLC

Problem:

In spite of adaptation to the NCU/PLC address in the mmc.ini file, no connection to the NCU/PLC can be established.

Possible cause:

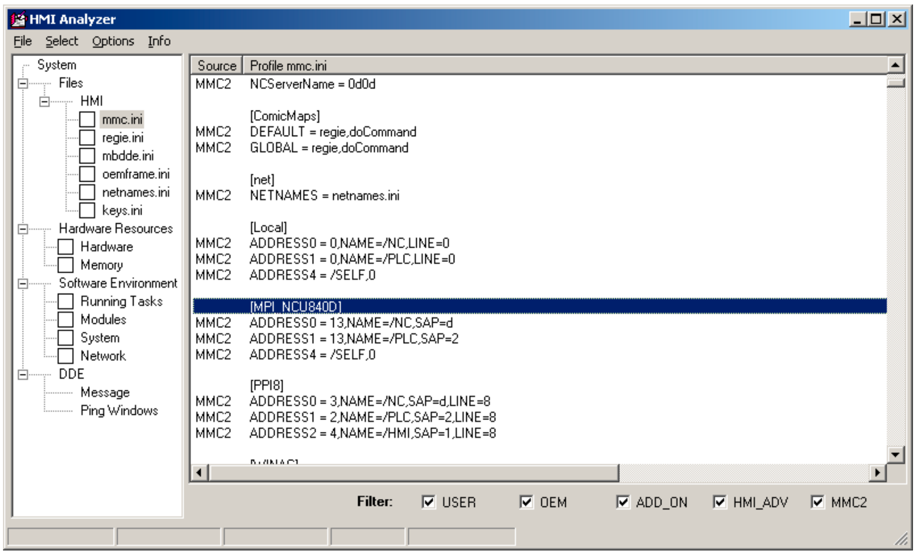
Due to the hierarchical execution of the HMI directory, the entry made in the mmc.ini file was overwritten by an entry with a higher priority and is therefore not in effect.

For example, an entry was made in the OEM directory:

```
[NCU840D]
ADDRESS0=5,NAME=/NC,SAP=d
ADDRESS1=6,NAME=/PLC,SAP=1
```

This is then overwritten in the user directory:

```
[NCU840D]
ADDRESS0=14,NAME=/NC,SAP=d
ADDRESS1=14,NAME=/PLC,MAX_BUB_CYCLICS=0
```



```
Source  Profile mmc.ini
MMC2    NCServerName = 0d0d

        [ComicMaps]
MMC2    DEFAULT = regie,doCommand
MMC2    GLOBAL = regie,doCommand

        [net]
MMC2    NETNAMES = netnames.ini

        [Local]
MMC2    ADDRESS0 = 0,NAME=/NC,LINE=0
MMC2    ADDRESS1 = 0,NAME=/PLC,LINE=0
MMC2    ADDRESS4 = /SELF,0

        [MPI_NCU840D]
MMC2    ADDRESS0 = 13,NAME=/NC,SAP=d
MMC2    ADDRESS1 = 13,NAME=/PLC,SAP=2
MMC2    ADDRESS4 = /SELF,0

        [PPI8]
MMC2    ADDRESS0 = 3,NAME=/NC,SAP=d,LINE=8
MMC2    ADDRESS1 = 2,NAME=/PLC,SAP=2,LINE=8
MMC2    ADDRESS2 = 4,NAME=/HMI,SAP=1,LINE=8
```

Filter: ☑ USER ☑ OEM ☑ ADD_ON ☑ HMI_ADV ☑ MMC2

Figure 5-12 Checking the connection to the control

HMI Analyzer:

Under Files → HMI, the mmc.ini is selected. The resulting mmc.ini file from all the HMI directories is now displayed (mmc2, hmi_adv, add_on, oem, user). Now it becomes clear whether the entry that was made is also truly active or was overwritten by another entry. In addition, it is possible to recognize which HMI directory the entry comes from.

(3) Blocking, very slow system

Problem:

The system is very slow and blocks the area switcher completely in some aspects. After a certain period of time, all the operating actions are suddenly processed sequentially.

Possible causes:

The area application that is newly selected by the area switcher sets up hotlinks when loading. When loading a form with DCTL, the message DDEInitialize is sent to the main window of all processes. If a main window is very slow to answer or does not answer this notification, then this can be the cause of the slow or blocked area switcher.

Rule:

"Top-level" windows must not be allowed to block the system for a long period of time.
DDE Clients may not use the DDEML. They must use the DCTL.

HMI Analyzer:

Select the DDE/Ping Windows item. Now you can check whether all windows answer the DDE broadcasts while the cyclic operation is being started. The "Answer Time" should be set to 100 ms and the cycle time of the query "Cyclic Time" to 1000 ms. The checkmark for Append should be set and the checkmark for ShowAll should be removed. The applications that appear in the list have not answered within the answer time. The displayed applications can be the reason for the slow, blocking system.

Note

Clients that establish DDE connections during operation are particularly affected. Clients that still use the DDEML should change to DCTL or limit the establishing of connections to the application start. The problem is only solved after the 32 Bit DCTL. VB4.0 programs must be ported accordingly.

For systems without much memory, the problem occurs more often. All of the processes exported to the hard disk must be loaded into the main memory (e.g. DDE initialize) and executed. This is very time-sensitive. Examples for time-critical applications are database accesses, complex computation programs, etc. Background processes, in particular, must be programmed in such a way that the applications that are running in the foreground are not blocked.

(4) Slow communication times

Problem:

If too many hotlinks are set up, the communication times are significantly slowed down.

Possible causes:

Depending on the load on the PLC, the number of hotlinks that are causing problems differs. Hotlinks represent a basic load for the control and as few as possible should be used.

HMI Analyzer:

The HMI Analyzer must be started prior to starting HMI-Advanced. This can be done in service mode. After HMI-Advanced has been booted, the HMI Analyzer displays the actually existing hotlinks under DDE → Message. Using the "Only Active Hotlinks" option, all active hotlinks can now be displayed. A check must now be made as to which data accesses will not need to be implemented as hotlinks but can also be implemented as a request. By doing this, the basic load on the control can be reduced.

Using the filter request, poke or execute, you can check whether these DDE functions are burdening the system. Cyclic requests with short cycle times burden the system. The cycle time can be determined from the "Time" column. This column displays the time that has elapsed since the start of windows in milliseconds.

Note

OPC DataAccess programmer

When programming an OPC Client Application, the variables to be read are automatically set up as a hotlink if a variable of the respective group is defined as a hotlink.

In addition, PLC variables for SINUMERIK OPC Server are automatically (Flag IsActive=True) set up as a hotlink.

Tool management

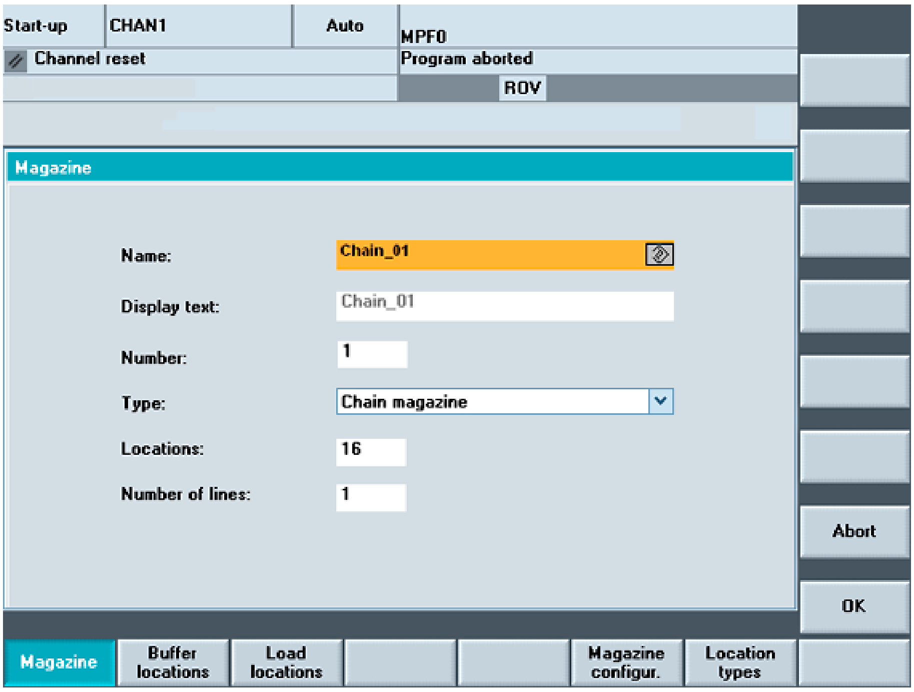
6

6.1 Setting tool management parameters on the HMI

6.1.1 How to create a real magazine

Creating real magazines

In this screen, magazines are defined with the appropriate data or displayed with existing data.



Name: Chain_01

Display text: Chain_01

Number: 1

Type: Chain magazine

Locations: 16

Number of lines: 1

Figure 6-1 Start-up: Magazines

Magazines	
Name	Enter a new name or select an existing name for the real magazine.
Display text	Language-dependent name of the magazine
Number	Display of the current magazine number
Type	Selection of a magazine type (chain magazine, revolver, box-type magazine)
Locations	Enter or display the number of magazine locations.
Number of columns	The "number of columns" is required for considering adjacent location and is only relevant for box-type magazines.

Note

Up to 32 magazines (including buffer and loading magazine) are possible, therefore a maximum of 30 real magazines.

Creating a new magazine

Follow the steps outlined below:

1. Press the "New" softkey. The magazine number is assigned by the system (in the order they were created).
2. Enter the magazine name with up to 32 characters in length. If it exists, the display text is immediately displayed from patm_xx.ini.
3. Select the magazine type:
   - Chain magazine
   - Revolver
   - Flat magazine
4. Enter the number of magazine locations.
5. For box magazines, the "number of columns" must be entered too.
6. Accept the data with the "OK" softkey.

Note

If the message "invalid value in magazine" appears, the number of locations and/or number of columns is incorrect. The "number of locations" value must be divisible by the "number of columns".

Example: 20 locations cannot be divided into 3 columns, but 21 locations can be.

Deleting a magazine:

1. Select the magazine name.
2. Press the "Delete" softkey.

   The magazine is deleted without a prompt.

Note

A magazine can only be deleted if it is not assigned to any magazine configuration.

Creating PLC data:

- When all magazines, buffers and loading points (for all channels / TO areas) are entered, the configuration data must be created for the PLC and transferred to it. To do this, press the "PLC data" softkey.

Note

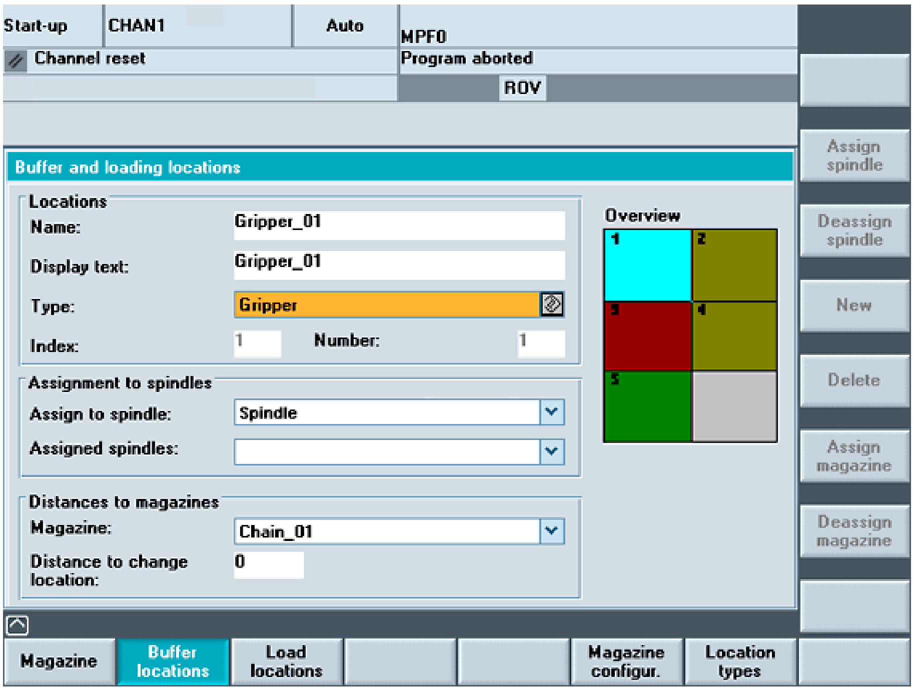
The next time the PLC is booted, the message "Deleting DBxx in PLC..." may appear. In this case, the specified DB is deleted via STEP 7. As an alternative, a suitable PLC archive can also be loaded.

6.1.2 How to parameterize a buffer

Buffer

Buffer locations are spindles, grippers, loaders and transfer locations. All buffer locations are managed in an internal buffer magazine with the number 9998.

In this screen, new buffer locations can be defined and existing ones displayed.



Start-up CHAN1 Auto MPF0
Channel reset
Program aborted
ROV

Buffer and loading locations

Locations
Name: Gripper_01
Display text: Gripper_01
Type: Gripper
Index: 1 Number: 1

Assignment to spindles
Assign to spindle: Spindle
Assigned spindles:

Distances to magazines
Magazine: Chain_01
Distance to change location: 0

Figure 6-2 Start-up: Buffer

Locations	
Name	Enter or select the name of the buffer.
Display text	Language-dependent name of the magazine If it exists, the display text is immediately displayed from patm_xx.ini.
Number	Display of the current magazine number
Type	Selection of buffer type (spindle/toolholder, gripper, transfer location, loader)
Index	The index counts the locations of a type.
Number	Display of the consecutive internal number under which the location is to be addressed
Overview	The number of all buffer locations is displayed as a graph. In addition, the currently selected buffer location (No.) is highlighted. Each "type" is displayed in a different color.

Note

The sequence must be adhered to when entering the buffer. The spindles must always be entered first. A number is assigned internally for each buffer location, via which the buffer location is addressed.

The buffer "spindle" with index 1 and spindle_1 in the NC have a direct relationship to one another. This means that for example the "spindle" buffer with index 1 must also be the 1st spindle for the NC, index 2 = 2nd spindle.

Creating a buffer location

Follow the steps outlined below:

1. Press the "New" softkey.
2. Enter a name: e.g. **Gripper_1**. If it exists, the display text is immediately displayed from patm_xx.ini.
3. Select the type: Transfer location, **gripper**, loader, spindle
4. The buffer location is created by pressing "OK", the location number and the index are internally assigned and incremented.

Assigning / deassigning a spindle:

1. The "Assign spindle" softkey assigns a buffer location (e.g. gripper) to a spindle. This informs the software of the mechanical relation between the gripper and spindle.
2. The "Deassign spindle" softkey cancels an existing spindle assignment.
3. Assigning/removing buffer locations to/from the magazines
4. When a magazine is selected, the distance to the change position must be entered. At least 0 must be entered, as otherwise the tool cannot be transported to this buffer location. The "Assign magazine" softkey assigns the buffer location to the magazine.
5. The "Deassign magazine" softkey cancels an existing assignment of a buffer location to a magazine.

Example

If, for example, tools are to be loaded from 2 magazines onto "Spindle_1", the assignment must be made for both magazines.

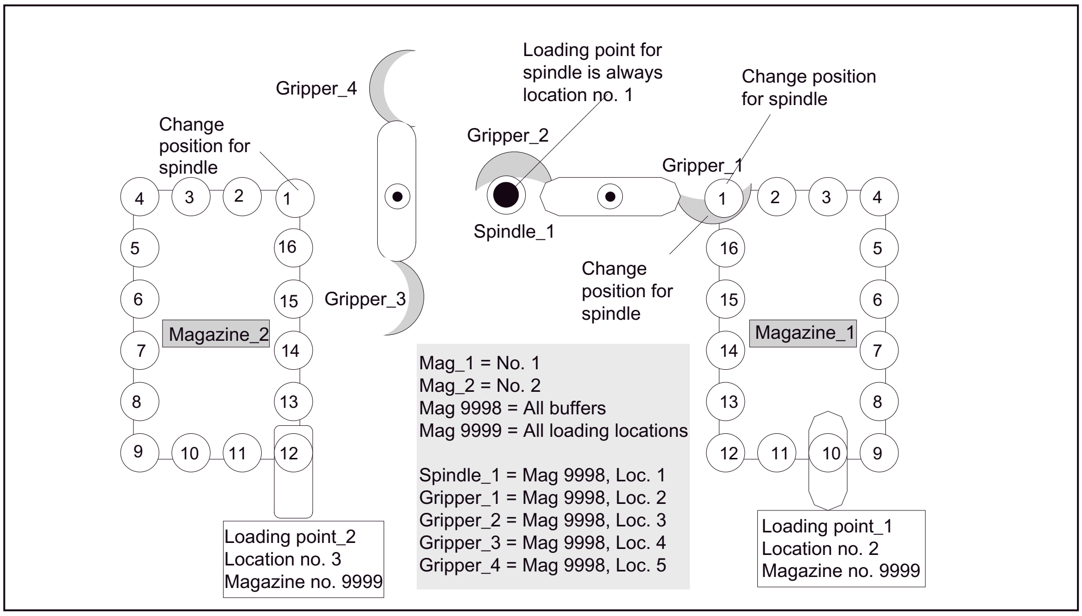

Gripper_4
Change position for spindle
Loading point for spindle is always location no. 1
Change position for spindle
Gripper_2
Gripper_1
Spindle_1
Change position for spindle
Gripper_3
Magazine_2
Magazine_1
4
3
2
1
5
16
6
15
7
14
8
13
9
10
11
12
1
2
3
4
16
5
15
6
14
7
13
8
12
11
10
9
Mag_1 = No. 1
Mag_2 = No. 2
Mag 9998 = All buffers
Mag 9999 = All loading locations
Spindle_1 = Mag 9998, Loc. 1
Gripper_1 = Mag 9998, Loc. 2
Gripper_2 = Mag 9998, Loc. 3
Gripper_3 = Mag 9998, Loc. 4
Gripper_4 = Mag 9998, Loc. 5
Loading point_2
Location no. 3
Magazine no. 9999
Loading point_1
Location no. 2
Magazine no. 9999

Figure 6-3 Example of a machine with buffer and loading magazine

No.	Name	Distance to change position
1	Spindle_1	Magazine_1, distance: 0 Magazine_2, distance: 0
2	Gripper_1	Magazine_1, distance: 0
3	Gripper_2	Magazine_1, distance: 0
4	Gripper_3	Magazine_2, distance: 0
5	Gripper_4	Magazine_2, distance: 0

See also

Language-dependence for user-defined name (Page 238)

6.1.3 Loading points and loading locations

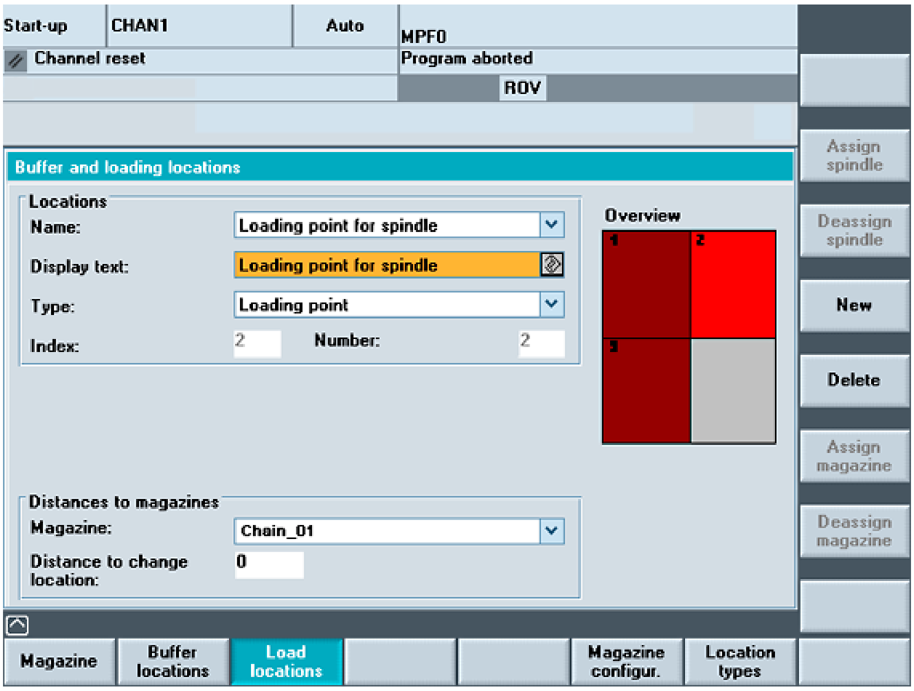
Loading locations

Loading locations are places that are needed for loading the magazine. There are two types of loading locations:

- Loading points
- Loading stations

All loading locations are stored in an internal loading magazine with the number 9999.

In this screen, new loading locations are defined or existing ones displayed.



Start-up | CHAN1 | Auto | MPF0
Channel reset | Program aborted
ROV

Buffer and loading locations

Locations
Name: Loading point for spindle
Display text: Loading point for spindle
Type: Loading point
Index: 2 Number: 2

Distances to magazines
Magazine: Chain_01
Distance to change location: 0

Figure 6-4 Start-up: Loading locations

Locations	
Name	Name of loading location (max. 32 characters).
Display text	Language-dependent name of the magazine. If it exists, the display text is immediately displayed from patm_xx.ini.
Number	Display of the current magazine number
Type	The options are loading point and loading station.
Index	The index counts the locations of a type.
Number	Display of consecutive internal number, under which the location is to be addressed
Overview	The overview displays the number of all buffers graphically. The numbers for the selected loading point/station are highlighted in color.

Note

Loading magazine 9999/1 is always automatically set for loading/unloading of the spindle (manual loading point).

Loading points

Loading points are areas on the machine at which it is possible to directly load the magazine, i.e. the tool can be directly inserted in the magazine at this point. The magazine location to be loaded is moved to the loading point here. For example, chain magazines have loading points.

The loading point is assigned to location type "7" ($TC_MPP1) in the magazine data.

Location type 7: If a tool is moved to this location from the magazine or toolholder, the NCK automatically removes the tool from this location when acknowledgement is received from the PLC.

Loading stations

A loading station is a location outside the magazine onto which the tool to be loaded is placed. The tool is then transported from that location to the magazine via a transport mechanism. Loading stations are generally used for box or chain magazines.

The loading station is assigned to location type "6" ($TC_MPP1) in the magazine data.

Location type 6: NCK does not differentiate between cases, i.e. if the tool was moved to this location, the tool remains there. It can only be removed (unloaded) from there through explicit operator action.

Sequence

Input of loading locations:

1. Press the "New" softkey.
2. Enter name: e.g. **Loading_1**. If it exists, the display text is immediately displayed from patm_xx.ini.
3. Select the type: e. g. **Loading point**
4. "OK" creates the loading point. The location number and the index are internally assigned and incremented.

Assigning/deassigning loading locations and magazines:

1. When a magazine is selected, the distance to the change position must be entered (at least 0).
2. You can now establish an assignment by pressing the "Assign spindle" softkey.
3. Distance 0 is usually used for the "manual loading point" (spindle loading point).

Example

Both magazines can be loaded via Spindle_1. Loading point_1 is only assigned to Magazine_1 and Loading point_2 is only assigned to Magazine_2.

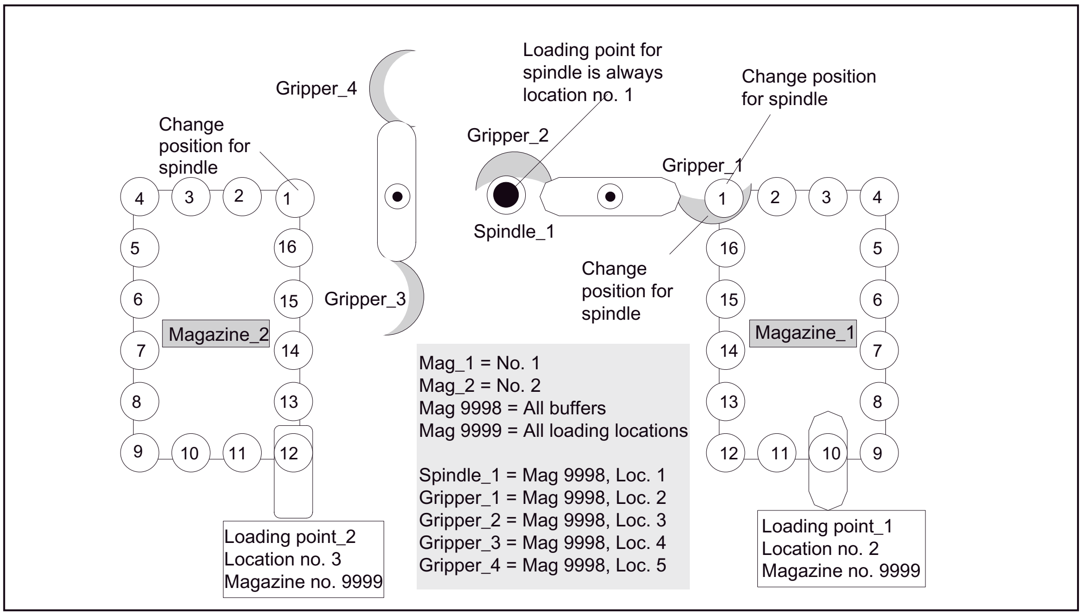

Gripper_4
Loading point for spindle is always location no. 1
Change position for spindle
Change position for spindle
Gripper_2
Gripper_1
Spindle_1
Change position for spindle
Gripper_3
Magazine_2
Magazine_1
4
3
2
1
5
16
6
15
7
14
8
13
9
10
11
12
1
2
3
4
16
5
15
6
14
7
13
8
12
11
10
9
Mag_1 = No. 1
Mag_2 = No. 2
Mag 9998 = All buffers
Mag 9999 = All loading locations
Spindle_1 = Mag 9998, Loc. 1
Gripper_1 = Mag 9998, Loc. 2
Gripper_2 = Mag 9998, Loc. 3
Gripper_3 = Mag 9998, Loc. 4
Gripper_4 = Mag 9998, Loc. 5
Loading point_2
Location no. 3
Magazine no. 9999
Loading point_1
Location no. 2
Magazine no. 9999

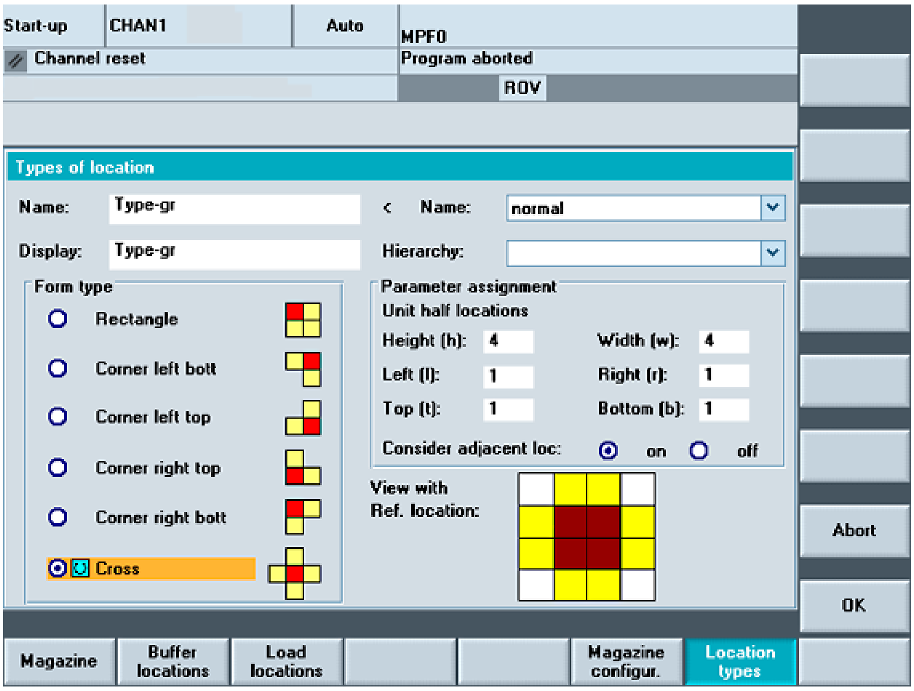
Figure 6-5 Example of a machine with 2 magazines and 3 loading locations

No.	Name	Distance to change position
1	Manual loading point (= spindle loading point)	Magazine_1, distance: 0 Magazine_2, distance: 0
2	Loading point_1	Magazine_1, distance: 9
3	Loading point_2	Magazine_2, distance: 11

6.1.4 Select location type

Location types

In this screen, new location types are defined with the required data or existing ones are displayed.



Start-up | CHAN1 | Auto | MPF0

Channel reset

Program aborted

ROV

Types of location

Name: Type-gr

Display: Type-gr

< Name: normal

Hierarchy:

Form type

○ Rectangle

○ Corner left bott

○ Corner left top

○ Corner right top

○ Corner right bott

● Cross

Parameter assignment

Unit half locations

Height (h): 4 Width (w): 4

Left (l): 1 Right (r): 1

Top (t): 1 Bottom (b): 1

Consider adjacent loc: ● on ○ off

View with Ref. location:

Figure 6-6 Start-up: Location types

Location types	
Name	Name of the location type (max. 32 characters).
Display	Language-dependent name of the magazine. If it exists, the display text is immediately displayed from patm_xx.ini.
Hierarchy	To overcome the inflexible classification of magazine locations according to location type, locations can be arranged in ascending order, i.e. in a hierarchy (see Section "Hierarchy of location types").
Form type	The position of the reference location (tool shank) is specified via the form type.
Parameterization	Definition of height and width as well as free half locations (left, right, top, bottom) (see examples).
Consider adjacent location	This information is stored magazine specifically (magazine configuration) and it relevant for the location search.

Assigning parameters for a location type

The number of half locations occupied by a tool in the magazine is defined when a location is parameterized. This corresponds to the tool size.

The four-digit number, e.g. 2 2 2 2, defines the half locations in the order left, right, top, bottom starting from a reference point. For setting the location type parameters, left plus right equals width and top plus bottom equals height.

The half locations that are not occupied are also defined using left, right, top and bottom (do not confuse this with the tool size!).

Reference location

The reference location is the physical location in the magazine. It is used as a reference point for specifying the tool size and is required for calculating the magazine assignment. The size of the reference location is always represented as tool size 1 1 1 1. (For parameter settings, refer to "Normal location type")

6.1.5 Example: Parameterizing location types

Normal location type

A tool which occupies one magazine location exactly has tool size 1 1 1 1. This tool is described as a "Normally sized tool".

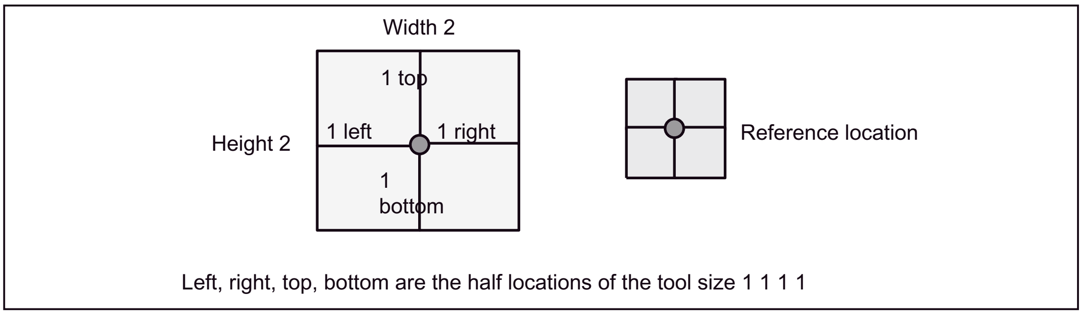

Width 2
Height 2
1 top
1 left
1 right
1 bottom
Reference location
Left, right, top, bottom are the half locations of the tool size 1 1 1 1

Figure 6-7 Normal location type

Parameter settings for the location type:

- height (h): 2, width (b): 2
- left (l): 0, right (r): 0
- top (t): 0, bottom (b): 0

Oversized location type for chain magazines

In a chain magazine, a tool with the size 2 2 1 1 occupies one half location on the right and one half location on the left in addition to the normal magazine location.

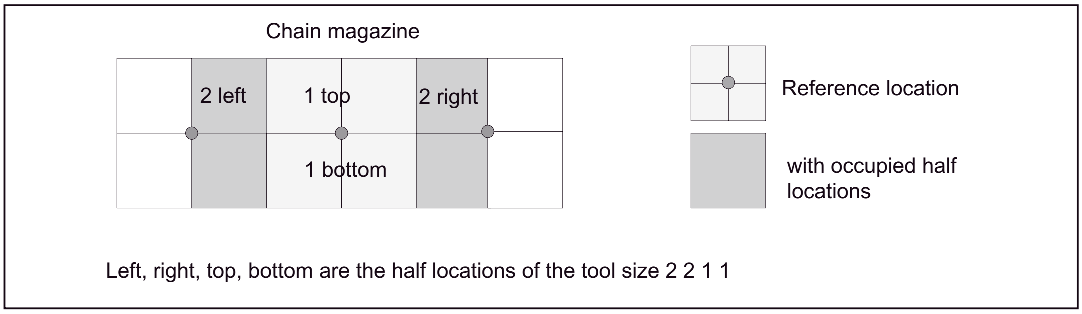

Chain magazine
2 left
1 top
2 right
1 bottom
Reference location
with occupied half locations
Left, right, top, bottom are the half locations of the tool size 2 2 1 1

Figure 6-8 Oversized location type for chain magazines

Parameter settings for the location type:

- height (h): 2, width (b): 4
- left (l): 0, right (r): 0
- top (t): 0, bottom (b): 0

Oversized location type for box magazines

In a box magazine, a tool with the size 2 2 2 2 occupies one half location in each direction in addition to the normal magazine location.

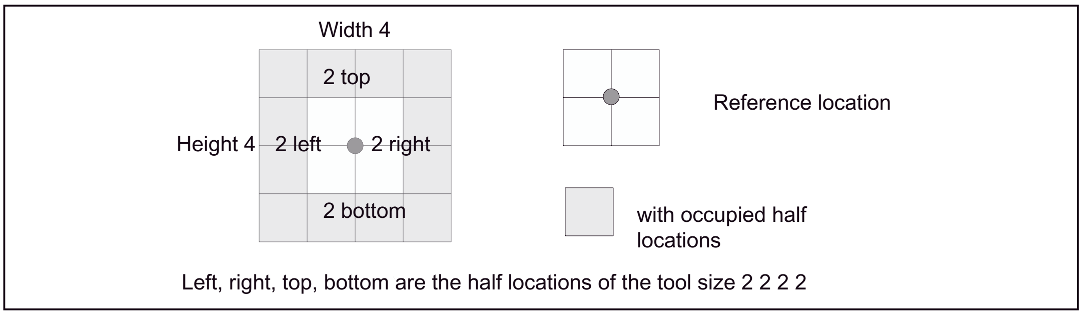

Width 4
2 top
Height 4
2 left
2 right
2 bottom
Reference location
with occupied half locations
Left, right, top, bottom are the half locations of the tool size 2 2 2 2

Figure 6-9 Oversized location type for box magazines

Parameter settings for the location type:

- height (h): 4, width (b): 4
- left (l): 0, right (r): 0
- top (t): 0, bottom (b): 0

Oversized location type with free half locations for box magazine

In a box magazine, a tool with the size 2 2 2 2 occupies one half location in each direction in addition to the normal magazine location.

However, with this location type the half location in each corner is not used.

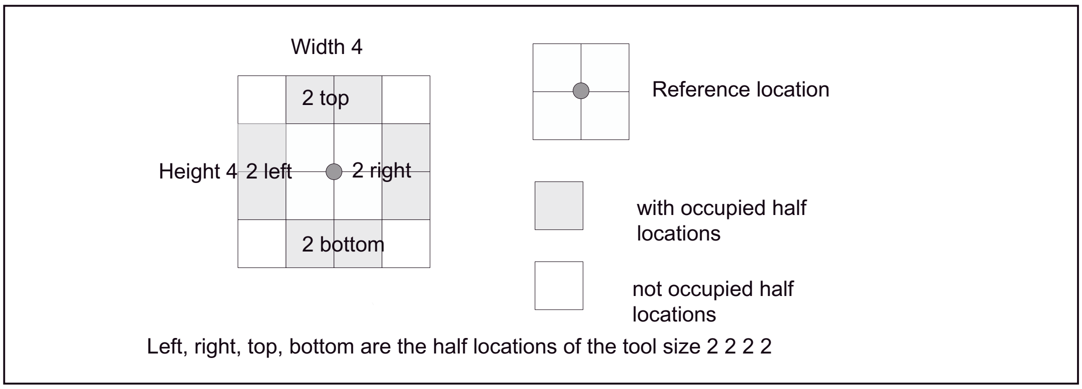

Width 4
2 top
Height 4
2 left
2 right
2 bottom
Reference location
with occupied half locations
not occupied half locations
Left, right, top, bottom are the half locations of the tool size 2 2 2 2

Figure 6-10 Oversized location type with free half locations for box magazine

Parameter settings for the location type:

- height (h): 4, width (b): 4
- left (l): 1, right (r): 1
- top (t): 1, bottom (b): 1

The half locations not occupied are defined by the parameters left, right, top, bottom as "FREE".

6.1.6 Example: To assign a hierarchy to location types

Hierarchy of location types

To overcome the inflexible classification of magazine locations according to location type, locations can be arranged in ascending order, i.e. in a hierarchy. Several such hierarchies can be set up for TO units but a location type can only belong to one hierarchy. Not allowed are: e.g. A < B and A < C or A < E and B < E.

This hierarchy ensures that a tool that only requires a "small" location type can also be placed in a "larger" location type if no "small" locations are free.

If a tool is to be inserted in the magazine, the location type decides which locations are available. If there is a hierarchy for this location, the locations are allocated in accordance with this hierarchy.

Application case

Let us assume, for example, that a tool of location type A is to be stored in the tool-holding magazine or a location search is to take place for a location of type A.

The following location hierarchy shall apply for this example: A < B < C.

First a check is performed to see whether there is a location with type A in the magazine to be searched. If there is not, the search function will proceed to search for a location of type B or C.

Example 1:

Existing location types: A, B, C

Hierarchy definitions: A < B, B < C

This gives an overall hierarchy of A < B < C

Example 2:

Existing location types: A, B, C, D, E

Hierarchy definitions: A < B, B < D, C < E

1st hierarchy: A < B < D

2. hierarchy: C< E

Sequence

Enter new location type:

1. Press the "New" softkey
2. Enter name (max. 32 characters). If it exists, the display text is immediately displayed from patm_xx.ini.
3. Select the form type
4. Enter the height and width of the location type in half locations. Depending on form type, enter left, right, top, bottom as required (unoccupied half locations).
5. Enable or disable "Consider adjacent location"
6. In the view graphic, set the position of the reference location using the cursor keys
7. Press the "OK" softkey to save

Delete location type:

- The "Delete" softkey deletes the selected location type. This is only possible if it has not been assigned to a magazine.

Create hierarchy:

1. Select small location type (name left) (in our example location type A).
2. Select larger location type (name right) (in our example location type B).
3. Press the "Create hierarchy" softkey to create the hierarchy. The hierarchy name in screen 1-9 corresponds to the name of the larger location type and is displayed in the hierarchy field (B in example).

Delete hierarchy:

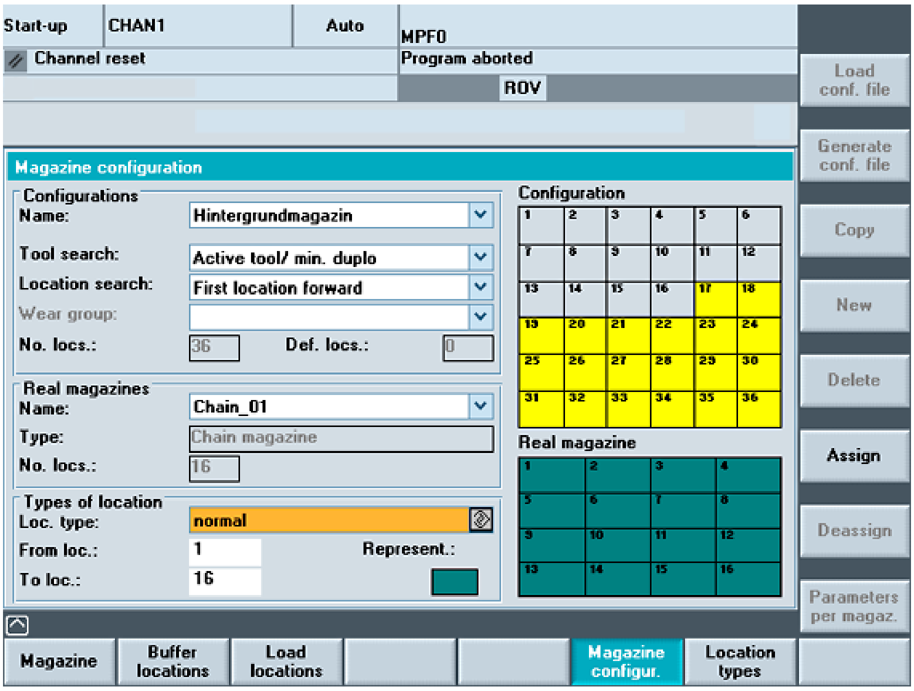
- You can remove the hierarchy selected in the "Hierarchy" field by pressing the "Delete hierarchy" softkey.

6.1.7 Magazine configuration

Create magazine configuration

For each TO unit, there is only one common magazine configuration for configuring the tool management. A magazine configuration can consist of one or more real magazines. As one TO unit can be assigned to several channels, this magazine configuration is available to the associated channels simultaneously.

In this screen, new data required for the selected magazine is defined or existing data is displayed.



Start-up | CHAN1 | Auto | MPF0
Channel reset | Program aborted
ROV

Magazine configuration

Configurations
Name: Hintergrundmagazin
Tool search: Active tool/ min. duplo
Location search: First location forward
Wear group:
No. locs.: 36 Def. locs.: 0

Real magazines
Name: Chain_01
Type: Chain magazine
No. locs.: 16

Types of location
Loc. type: normal
From loc.: 1
To loc.: 16
Represent.:

Figure 6-11 Magazine configuration

Configurations

Name: Name of the magazine configuration (max. 32 characters).

Tool search: transfer from configuration (Default setting: Setting for $TC_MAP10=0, NCK uses values from $TC_MAMP2).

active tool/min. duplo

Shortest path

active tool/min. $TC_TP10

Monitoring: min. actual value

Monitoring: max. actual value

Settings for the location coding can be variable (default) or fixed.

Exit the screen by pressing the "Cancel" or "OK" softkey.

Empty location search	First location forward / Current location forward
	Last location backward / Current location backward
	Symmetric current location
Wear group (wear group)	Leave tool status unchanged
	Actively change tool state
Number of locations:	Total number of locations in the configuration (all assigned magazines)
Def. locations:	Total number of locations in the configuration to which a location type was assigned

Real magazines

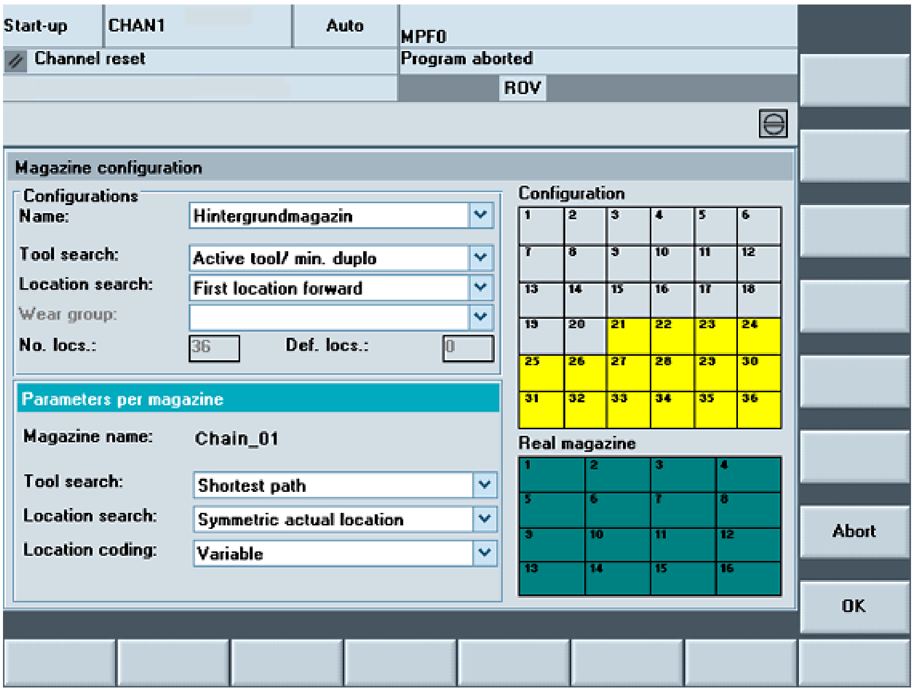
Name:	Name of the selected real magazine
Type:	Type of magazine
No. of Locations:	Number of locations for the selected magazine

Location types

Location type:	Name of selected location type
From location:	1. location to be defined
To location:	Last location to be defined

"Parameters per magazine" softkey:

The settings made in the previous screen ("Magazine configuration") become effective.



Magazine configuration

Configurations
Name: Hintergrundmagazin
Tool search: Active tool/ min. duplo
Location search: First location forward
Wear group:
No. locs.: 36 Def. locs.: 0

Parameters per magazine
Magazine name: Chain_01
Tool search: Shortest path
Location search: Symmetric actual location
Location coding: Variable

Configuration

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

Real magazine

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Figure 6-12 Set parameters for each magazine

6.1.8 How to create and load a configuration

Create new magazine configuration

The "Configuration" and "Real magazine" screens show the location type assignments for the entire configuration or the selected magazine.

1. Press the "New" softkey.
2. Enter a name, e.g. Example_documentation (32 characters max.)
3. Select selection menu for the tool search, location search and wear group and make the settings as appropriate.
4. Press the "OK" softkey (create the magazine configuration).

> **Note**
>
> The "Assign" and "Deassign" softkeys always refer to the selection field the cursor is positioned on:
>
> - Magazine
> - Location type

Process

Assign/deassign real magazines:

1. Select a real magazine and press the "Assign" softkey. This real magazine is then included in the magazine configuration.
2. After each assignment, the total number of magazine locations is accordingly updated in the configuration.
3. You can remove a magazine again from the magazine configuration by pressing the "Deassign" softkey.

Assign/deassign location types

1. Select location type. Enter magazine locations to which this location type is to be assigned.
2. Example: "From location: "1", "To location: "10".
3. Press the "Assign" softkey. The defined locations are displayed with the color for this location type
4. You can remove the assignment made for a location type by pressing the "Deassign" softkey.

Create configuration file:

- Press the "Create config. file" softkey. An INI file is created which can later be uploaded to the NCK.

Copy configuration

1. Press the "Copy" softkey.
2. Enter a new name and confirm by pressing the "OK" softkey.

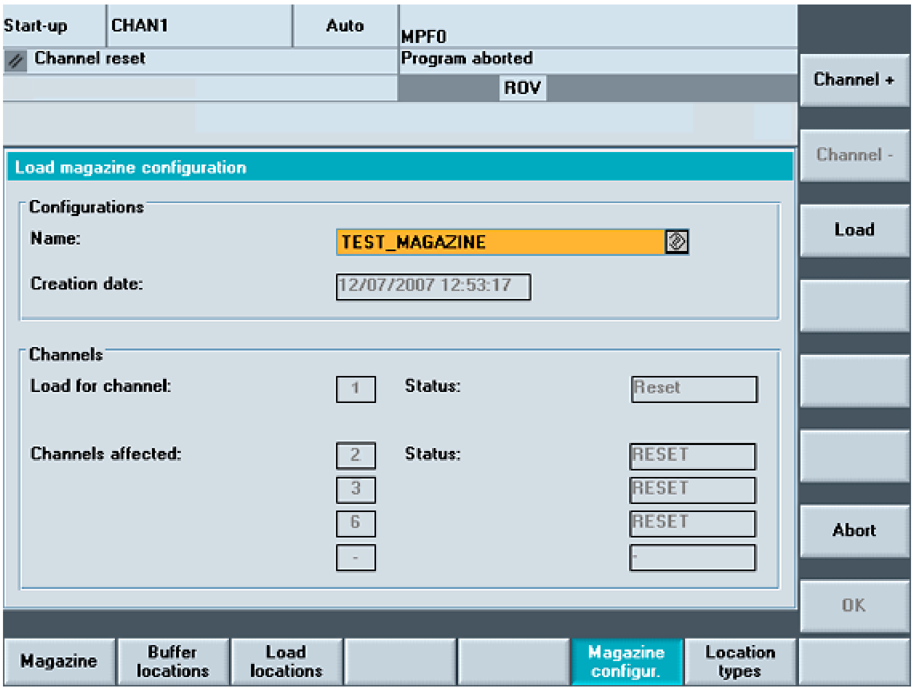
Delete magazine configuration:

1. If the cursor is positioned on the selection field for the configuration, you can delete the selected configuration by pressing the "Delete" softkey.
2. Load magazine configuration
3. Activating the "Load config. file" softkey opens the "Load magazine configuration" screen 1-16.

Load magazine configuration:

- Here, the previously created INI file is uploaded to the NCK via the "Load" softkey. The procedure is channel-specific; only one configuration is possible for each TO unit.

  (This means if TO unit 1 is assigned to channels 1, 2, 3 and 6, the configuration which was loaded in channel 1 is automatically valid for channels 2, 3 and 6 too.)



Start-up CHAN1 Auto MPF0
Channel reset
Program aborted
ROV

Load magazine configuration

Configurations
Name: TEST_MAGAZINE
Creation date: 12/07/2007 12:53:17

Channels
Load for channel: 1 Status: Reset
Channels affected: 2 Status: RESET
3 RESET
6 RESET
- -

Figure 6-13 Load configuration file

6.2 Settings in the configuration file

6.2.1 Parameterize the paramtm.ini file

Configuration

Customized differences from the factory-set ini data are stored in the "user" or "oem" directories. Only changes to the visual aspect of the operator interface are stored here, i.e. changes that can be made by settings in the HMI operator interface itself.

All data that describes the operator interface of the tool manager are saved in the ...\user\paramtm.ini file. To edit the file, select "Start-up" → "HMI" → "Editor". The paramtm.ini file can also be created on an external PC and copied into the ..\user directory.

Country-specific sections are parameterized in the "language\patm_xx.ini" file in the [BatchTools] section. "xx" stands for the national identifier, e.g. gr for German, uk for English. Other languages: see Table 2-2.

Additional functions

The following additional functions have been implemented in the magazine, tool, work correction list and tool details:

- Parameter settings for bitmaps in the list
- Tool identifiers and duplo numbers can be changed in the lists
- New magazine list with several lines
- Job processing of tools
- Tool-status bit "Pre-warning limit" can be changed in displays of lists
- Tool status bit "Unload detection" and "Load detection" in displays of lists and details can be changed
- New tool types:

  550 Steel profile

  700 Slotting saw

  711 Edge probe

  720 oriented probe

  730 Stop

- If adapter data is set in the NCK, the magazine list can be displayed either transformed or not transformed (softkey on the ETC bar). The setting is made in paramtm.ini, section [TMMODES], with the entry START_MAGLIST_TRANSFORMED
- Suppression of status bits for tool cabinet, code carrier, SINCOM excerpt from paramtm.ini:
  - ;Tool status:

    If a tool is removed from the NCK and transferred to an external medium (tool cabinet, code carrier, SINCOM), you can use the following screens to specify which tool status bits should be saved.
  - ;Code carrier:

    Since the standard conversion file wkonvert.txt has entered 1 byte for the tool status and a max. of 92 have been written to the code carrier up to now, CODECARRIER_TOOLSTATE_MASK receives the default value 92. If the value for CODECARRIER_TOOLSTATE_MASK is expanded, then the size of the dialog variables T9 in wkonvert.txt has to be adapted accordingly.

See also

Start-up of code carrier (Page 258)

6.2.2 Structure of the paramtm.ini file

Directory

Directory ...\user\paramtm.ini

Note

Until now, the mmc2\paramtm.ini parameter settings for tool management only contained comments consisting of documentation of the individual entries. Almost all comments had to be removed, as new entries caused the critical file length limit of around 63KB to be exceeded.

A version of paramtm.ini with comments can now be found in mmc2\paramtm.txt.

Overshooting and other errors when reading the parameterization are logged as before in the ...\user\paramtm.out file.

A comment can be inserted at the end of entries using a semicolon ";".

Structure of the file

[ACCESSLEVEL]

```
SKAVTIVTM=7
; Activates TM in the PARAM application
SKMGLIST=7
; Displays the magazine list
```

```
SKTLLIST=7
; Displays the tool list
SKACLIST=7
; Displays the tool offset list
SKTOOLLOAD=5
; Loading tools
SKTOOLUNLOAD=5
; Unloading tools
SKTOOLMOVE=7
; Moving tools in the magazine
SKSETTINGS=4
; Softkey settings
SKFILFCT=4
; Softkey file functions
SKNXTCHAN=7
; Enable Next Channel softkey
SKMAGCONF=4
; Configuration of magazines
SKTOOLCAT=7
; Tool catalog
SKTOOLCAB=7
; Tool cabinet
SKSINCOMLD=7
; Load tool from SINCOM (if code carrier is installed)
SKCTORSINCOM=7
; Code carrier functions or tool loading from SINCOM
; (if code carrier is installed)
SKMGLREPR1=7
; Display 1_MagList
SKMGLREPR2=5
; Display 2_MagList
SKMGLREPR3=5
; Display 3_MagList
SKNCTOOLDATA=7
; Read tool data from NC or file management
; (if ACTIVATE_EDGE_MANAGEMENT_IN_LISTS=True)
SKNCTOOLED=7
; Cutting edge data
SKNCTOOLSUPV=7
; Supervision data
SKNCTOOLDL=7
; DL data
SKNCTOOLGRIND=7
; Grinding parameters
SKNCDETAILS=7
; Read tool data from NC
; (if ACTIVATE_EDGE_MANAGEMENT_IN_LISTS=True)
```

```
SKNCNEWTOOLED=6
; Create new cutting edges in NC
SKNCDELTOOLED=6
; Delete cutting edges in NC
SKNCDELTOOL=5
; Delete tools in NC
SKTRAFO=7
; Toggle transformed / not transformed view of edge data
SKCHECKACTIVATE=6
; D check and activation
SKMGBUFFER=7
; Display the buffer
SKMGFIND=7
; Softkey search and position
Positioning
SKMGLISTPOS=7
; Softkey next magazine
SKMGNEXT=7
; Create tools in NC
SKTLNEWTOOL=6
; Display 1_ToolList
SKTLLREPR1=7
; Selection of Display 2_ToolList
SKTLLREPR2=5
; Display 3_ToolList
SKTLLREPR3=5
; Location search, user-defined 1
SKFINDPL1=7
; Location search, user-defined 2
SKFINDPL2=7
; Location search, user-defined 3
SKFINDPL3=7
; Location search, user-defined 4
SKFINDPL4=7
; Location search
SKFINDPL=7
; Location search, tool loading list
SKFINDPLACE=7
; Allows softkey current location
SKACTPLACE=7
; Softkey tool data in status loaded tools
SKLDTOOLDAT=7
; Load a magazine configuration
SKCONFLOAD=4
; Display 1_ActList
SKACLREPR1=7
; Display 2_ActList
```

```
SKACLREPR2=7
; Display 3_ActList
SKACLREPR3=7
; Softkey delete D numbers
SKDZERO=7
; Softkey search for D numbers
SKDFIND=7
; Softkey filter lists
SKBATCH=7
SKBFILTER1=7
; Softkey Filter1
SKBFILTER2=7
; Softkey Filter2
SKBFILTER3=7
; Softkey Filter3
SKBFILTER4=7
; Softkey Filter4
SKBFILTER5=7
; Softkey Filter5
SKBFILTER6=7
; Softkey Filter6
SKBMAGFILTER=7
; Softkey magazine selection (for filter)
SKBATREACT=7
; Softkey batch function "Reactivate"
SKBATTOCABIN=7
; Softkey batch function "In cabinet"
SKBATDELTOOL=7
; Softkey batch function "Delete"
SKBATUNLOAD=7
; Softkey batch function "Unload"
SKBFILTERACT=7
; Softkey batch function "Update filter"
SKBATLOAD=7
; Softkey batch function "Load"
SKBATLIST=7
; Softkeys for controlling the job processing
ChangeToolTypeWithoutConfirmation=-1
ChangeToolSizeAndToolplace_spec=-1
READ_GUD_LUD=7
WRITE_ZOA=7
READ_SYSVAR=7
EDIT_VIEW=7
```

[DETAILS]

```
TOOLBASE_COL1=20
TOOLBASE_COL2=20
```

```
TOOLBASE_COL3=20
TOOLBASE_COL4=20
```

[DEFAULT SETTINGS]

```
; Create default tool settings:
; see paramtm.ini or paramtm.txt in path mmc2
; Magazine list: Load, data input directly in the list:
; 0=Due to the missing input, the default settings must be
; confirmed with the "Tool data" screen if they
; are required.
; 1=The default settings become effective without confirmation
; (with the exception of tool identifier no.)
; 2=The default settings become effective without confirmation
; (including the tool identifier no.)
DEFAULT_WITHOUT_CONFIRM=0
; Half locations: From 1 to 7
TOOLSIZE_LEFT=1
; Half locations: From 1 to 7
TOOLSIZE_RIGHT=1
; Half locations: From 1 to 7
TOOLSIZE_UPPER=1
; Half locations: From 1 to 7
TOOLSIZE_DOWN=1
; Tool type, from 100 to 1000
TOOLTYPE=120
; Duplo number: From 1 to 32000
TOOLDUPLO=1
; Ident. no.: Max. length 27
TOOLIDENT=NEW
; Additive values, (default: 0):
; 1=Active tool
: 2=Permitted
; 4=Blocked
; 8=Measured
; 16=Warning limit reached
; 32=Being changed
; 64=Fixed location coding
; 128=Already in use
; 256=Tool in buffer
; 512=Blocked, not taken into account (because of PLC)
; 1024=Out (unload)
; 2048=In (loaded)
; 4096=Standard tool (constantly in the NCK)
; 8192=
; 16384=
TOOLSTATE=0
; Index of a defined location type
```

```
TOOLPLACESPEC=1
; 0=No monitoring (default)
; 1=Time monitoring
; 2=Workpiece count monitoring
TOOLMONITOR_MODE=0
; Tool search, number of replacement tool ($TC_TP10)
; 0 ... 32000
TOOLSEARCH_MODE=0
; Default setting of grinding-specific tool data at creation:
; If the machine is operating with inch/mm
; ($MN_CONVERT_SCALING_SYSTEM=1) conversion,
; the length unit must be specified.
; The following default values (TOOLGRIND..., if affected
; by the length unit) are specified on the basis
; of this length unit:
; 0 = mm (default)
; 1 = inch
TOOLGRIND_Default_Length_Unit=0
;Spindle number (as $TC_TPG1)
TOOLGRINDspinNoDress=1
;Chain rule (as $TC_TPG2)
TOOLGRINDconnectPar=1050629
;1050629 binary: 0000 0000 0001 0000 0000 1000 0000 0101
;Bit0=1 = Type
;Bit2=1 = Geo-L1
;Bit11=1 = Wear-L1
;Bit20 =1 = Base-L1
;Minimum wheel radius (as $TC_TPG3)
TOOLGRINDminToolRadius=0
;Minimum wheel width (as $TC_TPG4)
TOOLGRINDminToolWide=0
;Current width of grinding wheel (as $TC_TPG5)
TOOLGRINDactToolWide=0
;Maximum grinding wheel speed (as $TC_TPG6)
TOOLGRINDmaxRotSpeed=0
;Maximum grinding wheel peripheral speed (as $TC_TPG7)
TOOLGRINDmaxTipSpeed=0
;Inclination angle of inclined wheel (as $TC_TPG8)
TOOLGRINDinclAngle=0
;Compensation parameter for grinding wheel peripheral speed (as
$TC_TPG9)
TOOLGRINDparamNrCCV=3
```

[TMMODES]

```
; 0=Do not delete tool automatically
; when it is unloaded (magazine list only).
; 1=Delete tool automatically when it is unloaded
; (magazine list only)
; DELETE_TOOL_ON_UNLOAD=0
; 0=Do not process cutting edge parameters outside the tool type
; (default)
; 1=Process cutting edge parameters outside the tool type
; (if not equal to 0)
EDGE_PARAMS_OUT_OF_TOOLTYPE=1
; 0=display: Tool size left, right, top, bottom (default)
; 1=display: left, right
SHOW_TOOLSIZE_ONLY_LEFT_RIGHT=0
; Tool size display:
; True=display (default)
; False=do not display
; is only used if SHOW_TOOLSIZE_ONLY_LEFT_RIGHT = 0
; (or default)
SHOW_TOOLSIZE_COMPONENTS=left:=True, right:=True, top:=True,
bottom:=True
; The function "Activate D check" refers to:
; -1=all magazines with distance relationship to spindle/toolholder
; (default)
; 1=current magazine only
DCHECK_ACTIVATE=-1
; The "Activate D check" function can be carried out automatically
; when the working offset list is opened
; False=function can only be activated via softkey (default)
; True=function is automatically carried out when the working offset
; list is opened
DCHECK_AUTO_ACTIVATE=False
; DCHECK_AUTO_ACTIVATE_MODE is only used
; if DCHECK_AUTO_ACTIVATE=True
; 0=The "Activate D check" function is carried out automatically
; when the working offset list is opened, even in the main screen
, (contains "R parameters" softkey) (default)
; 1=The "Activate D check" function is carried out automatically
; when the working offset list is opened, but not in the
; main screen (contains "R parameters" softkey). Same behavior as
before
; implementation of DCHECK_AUTO_ACTIVATE_MODE
; (prior to HMI Advanced 06.03.19).
; Buffer location display:
; DB: Original name from the magazine configuration from the
database
; no locale-dependent texts
; DLL: Name = text from the language-DLL + index (default)
; Example: Spindle1, language-specific texts
NameOfBufferPlaceFrom=DB
; To display the "Create cutting edge" and "Delete cutting edge"
functions
; (only possible with multi-line display) in the
; "Tool / magazine list" main menu, the "Tool details" softkey
; can be replaced by the new "Data management" softkey.
; The "Tool details" softkey is then stored with the same
; functions behind the "Data management" softkey.
; False="Tool details" remains active (default)
; True="Data management" is activated
ACTIVATE_EDGE_MANAGEMENT_IN_LISTS=False
; If adapter data is set in the NCK, then the
; magazine list can be displayed as either a transformed or
untransformed
; list (softkey in ETC menu). The display mode can
; be preset for the boot up of the HMI.
; False = not transformed (default)
; True = transformed
; START_MAGLIST_TRANSFORMED=False
; The "Tool management" softkey in the "Parameters" main menu
; can be used to branch to the list below.
; 0 = standard list depending on the respective NCK version /
; machine data (default)
; 1 = Magazine list
; 2 = Tool list
; 3 = Working offset list
START_LIST=0
; The inch/metric setting is considered for the code carrier
; !! NOTICE: The settings described here are only
; relevant if the MD $MN_CONVERT_SCALING_SYSTEM=1 is set
; in the NC.
; The entry for DATABASE_LENGTH_UNIT is only analyzed if
; no unit was entered in the tool database,
; i.e. once under normal circumstances.
; If it is an old NC without an inch/metric conversion
; function, or if $MN_CONVERT_SCALING_SYSTEM=0,
; then the NC operates without inch/metric conversion. Therefore
; no conversions are carried out in relation to the code carrier
either.
; 1 = No definition for inch/metric in relation to code carrier
(default).
; The setting $MN_CONVERT_SCALING_SYSTEM=1 in the NC means
; that inch/metric conversion is to be used.
; Therefore the user must determine the unit of the data
; on the code carrier or to be written to the code carrier.
; As with setting -1 this does not take place, all softkeys
; for code carrier activities are disabled.
; 0 = mm. It is assumed that the relevant data has been or is to be
; stored on the code carrier in mm.
. If "inch" is set in the NCK
; all softkeys are disabled and code carrier functions start.
; 1 = inch. It is assumed that the relevant data has been or is to
be
```

```
; stored on the code carrier in inches.
. If "inch" is set in the NCK
; all softkeys are disabled and code carrier functions start.
DATABASE_LENGTH_UNIT=-1
; The inch/metric setting is considered for the code carrier
; !! NOTICE: The settings described here are only
; relevant if the MD $MN_CONVERT_SCALING_SYSTEM=1 is set
; in the NC.
; If it is an old NC without an inch/metric conversion
; function, or if $MN_CONVERT_SCALING_SYSTEM=0,
; then the NC operates without inch/metric conversion. Therefore
; no conversions are carried out in relation to the code carrier
either.
; -1 = inch/metric is ignored (default). Data traffic
; between code carrier and NCK/HMI takes place without consideration
; of inch/metric Behavior as up to now.
; 0 = mm. In future all relevant data is written in mm
; to the code carrier.
; For future data transfers between the code carrier and NC
; this is taken into account accordingly.
; 1 = inch. In future all relevant data is written in inches
; to the code carrier.
; For future data transfers between the code carrier and NC
; this is taken into account accordingly.
CODECARRIER_LENGTH_UNIT=-1
; Tool status: If a tool is removed from the NCK and transferred
; to an "external" medium (tool cabinet, code carrier, SINCOM),
; then you can use the following screens to specify which
; tool status bits should be saved.
; Code carrier: Since the standard wkonvert.txt file has entered 1
byte
; for the tool status and a max. of 92 have been written to the code
carrier
; up to now, CODECARRIER_TOOLSTATE_MASK receives the
; default value 92.
; If the value for CODECARRIER_TOOLSTATE_MASK is expanded,
; then the size of the dialog variables T9 in wkonvert.txt
; has to be adapted accordingly.
; 1=Active Tool
; 2=Allowed
; 4=Disabled
; 8=Measured
; 16=Warning limit reached
; 32=In change
; 64=Fixed place coding
; 128=Was used
; 256=Tool in buffer
; 512=Disabled, ignored (because of PLC)
; 1024=Out (unload)
; 2048=in (Load)
; 4096=Regular tool (permanent in NCK)
; 8192=
; 16384=
; Default is 4828 (4+8+16+64+128+512+4096),
; For code carrier 92 (4+8+16+64)
CABIB_TOOLSTATE_MASK=4828
SINCOM_TOOLSTATE_MASK=4828
CODECARRIER_TOOLSTATE_MASK=92
TOOLSEARCH_TC_TP10_FROM_DB=0
; Tool search:Processing $TC_TP10 in conjunction with the
; tool cabinet/catalog
; 0 = (Default) If a tool is moved from the tool cabinet into
; the NCK, the value "Spare tool no. ($TC_TP10)"
; is not transferred to the NCK.
; ($TC_TP10) is set in the NCK to 0.
; 1 = The value "Spare tool no. ($TC_TP10)" is transferred
; from the tool cabinet into the NCK and displayed in the
; tool catalog/tool cabinet screens.
```

[General]

```
; Settings for "Write current data for tool management
; HMI to NCDDE variables on changing to
; WIZARD screens or on pressing WIZARD softkeys":
; All settings must be set in one single line by the name
; parameter.
; An option is enabled by the value "True" and
; disabled by the value "False" or by the absence of the
; name parameter in the line.
; "EnableAllTogetherWriteToNcdde := True": all data in a single
; NCDDE variable
; "EnableSingleWriteToNcdde := True": each data item has a separate
; NCDDE variable
; Both settings can be simultaneously active.
; If neither option is active, the NCDDE variables are not
; written to.
; "WriteChangesWhenStateChanged := True": data is written when any
; softkey is activated, not just WIZARD softkeys.
HMICurDataInterface = EnableAllTogetherWriteToNcdde := True,
EnableSingleWriteToNcdde := True,
WriteChangesWhenStateChanged := False
; $MM_WRITE_TOA_FINE_LIMIT and $MM_USER_CLASS_WRITE_FINE are applied
; to the geometry values and basic values of cutting edge data
UseFineLimitForToolGeoAndAdapt=False ; (default)
;UseFineLimitForToolGeoAndAdapt=True
; 1: Read language-specific INI files (language\patm_xx.ini)
; (default)
; 0: Do not read
ReadLanguageIni=1
SearchPlaceMethod=NoInternalTool ; Don’t modify!
; Tool details forms: colors for mixed adapter transformed/
; untransformed display:
; Hex values, 8 characters per color(SSBBGGRR where SS=System,
; BB=Blue, GG=Green, RR=Red)
; 4 Values for:
TransformedText, TransformedBackground,
NotTransformedText, NotTransformedBackground
DetailsMixedTrafoColors=WinTxt, li-brown,WinTxt ,li-blue
DetailsMixedTrafoColors =80000008,00008080,80000008,00FFFF00
; Allow display and edit of all 3 len parameters L1 L2 L3 in
; cut edge geo, cut edge wear, SC, EC independent of tool type and
; independent of $SC_TOOL_LENGTH_CONST and $SC_TOOL_LENGTH_TYPE
; in tool management lists, detail forms, cabin and catalogue.
AllwaysAllowL1L2L3IO=False ;default
; AllwaysAllowL1L2L3IO=True
; Allow change of Tool state bit 8 (least significant bit is Bit 1)
; (TC_TP8_8 "ToolState Used, Tool was being used") via GUI (list
; and details forms) for NCU tools.
; default: FALSE
AllowChangeOfTC_TP8_8 = False ;default
; AllowChangeOfTC_TP8_8 = True
```

[GeneralSettingsForMagAndToolList]

```
ToolState_Lang_12345678=<Empty> ; use language-DLL
; If MagPlaceState_Lang_12345678 and ToolState_Lang_12345678
; are not defined here or equal "<Empty>", then the
; locale-dependent values in the magazine and tool list from
pa_xx.dll are
; displayed.
; If values are set here and the entry in the "[General]" section
for
; "ReadLanguageIni" equals 1, then the files
; mmc2\language\patm_gr.ini, user\language\patm_gr.ini etc. are
searched
; in the same section as here. The name for the entry that is used
; in the locale-dependent file is the value of the entry
; from paramtm.ini.
; If an entry is found in the locale-dependent file,
; then it is used as text.
; If it is not found or if "...=<Empty>" is found,
; then the value from the paramtm.ini file is used as text.
; The 8 characters in MagPlaceState_Lang_12345678 and
; ToolState_Lang_12345678 correspond to the 8 magazine location
; and tool states and are displayed as location or tool status
values
```

```
; in the magazine list and in the tool list.
; Example: ToolState_Lang_12345678=12345678_ToolState_Lang
MagPlaceState_Lang_12345678=12345678_MagPlaceState_Lang
; use patm_*.ini
; The new magazine location status bits can be displayed in each
; list display. The parameters are set in the sections:
"[1_MagList]", "[2_MagList]", "[3_MagList]"
"[1_ToolList]", "[2_ToolList]", "[3_ToolList]"
"[1_ActList]", "[2_ActList]", "[3_ActList]"
; Entries are for example:
12=TC_MPP4_9, 1, TC_MPP4_9 ;PlaceStatus Left,
;Reserved in left half location
13=TC_MPP4_10, 1, TC_MPP4_10 ;PlaceStatus Right,
;Reserved in right half location
14=TC_MPP4_11, 1, TC_MPP4_11 ;PlaceStatus Top,
;Reserved in upper half location
15=TC_MPP4_12, 1, TC_MPP4_12 ;PlaceStatus Bottom,
;Reserved in lower half location
16=TC_MPP4_13, 1, TC_MPP4_13 ;PlaceStatus Bit 13 of 1 to 16
17=TC_MPP4_14, 1, TC_MPP4_14 ;PlaceStatus Bit 14 of 1 to 16
18=TC_MPP4_15, 1, TC_MPP4_15 ;PlaceStatus Bit 15 of 1 to 16
19=TC_MPP4_16, 1, TC_MPP4_16 ;PlaceStatus Bit 16 of 1 to 16
; Example: MagPlaceState_Lang_12345678=12345678_MagPlaceState_Lang
; MagPlaceState_Lang_12345678=<Empty>
; For alphanumeric columns in a list:
; Width of a character in "twips". The approximate column width
; is calculated by multiplying the value entered here
; by the number of characters from the column parameter setting
ColumnWidthTwipsPerAlphaCharacter=140
; For numerical columns in a list:
; Width of a character in "twips". The approximate column width is
; calculated by multiplying the value entered here by
; the number of characters from the column parameter setting
ColumnWidthTwipsPerNumericCharacter=100
; Number of data elements in the magazine list or in the
; tool list at each internal data scan.
; Range: 1 to 27, default 18.
; As of version P4.3.8.the data-scanning rate has been
; improved.
; While internal data scanning is taking place, there is no
; response to softkey actuation. This time should therefore
; not exceed 1 second.
; The value from "NumLinesPerReq" is used to scan data from a
; complete list in the background, after a list has been
; selected by pressing a softkey or after startup.
; The number of visible lines in the list is used to retrieve data
; if the data display is refreshed after changes to
; the data or on scrolling through the list and the
; number of visible lines in the list is used to smaller than
; NumLinesPerReq. If data exchange between the HMI and NCK
; proceeds slowly (with the NCU 810 D), this value should be
; reduced to 17 in order to achieve a response time of
; around 1 s when retrieving data from a complete list
; in the background. The setting applies to all lists
; for which no individual settings have been made.
; Individual settings are useful if a list contains a large
; number of columns.
; In this case the value should be lowered to about 10 or 5.
; Too many columns in a list that is in frequent use
; is not recommended, because the time needed to retrieve
; data for the entire list would be too long and the user would have
to
; wait a long time for the display.
; To set an individual list to individual setting, add the following
line
; to the section (e.g. [2_ToolList]) of the corresponding list:
; "NumLinesPerReq".
; Use 27 for NumLinesPerReq=27 from Version P4.3.8!
; Up until P4.3.8 the value 7 worked well.
; Width of the bitmap image for the current tool and the
; current tool magazine location in the list displays.
; Unit: number of characters; the width of a character is
; set by "ColumnWidthTwipsPerAlphaCharacter" or
; "ColumnWidthTwipsPerNumericCharacter". Lowest value: 1,
; highest value: 32, default: 5
; User-defined images can also be used:
; if the file name is specified without a path or with the path
MMC2,
; then the system searches for the bitmap file in the directories
"user", "oem",
; "add_on" and "mmc2". ; The first hit is then
; used. It is not advisable to generate user-defined bitmaps
; that are too large. The ratio of width to height should
; roughly correspond to the display in the lists, to avoid
; distorting the display.
; Use WidthOfActBitmapsInCharacters = 7, if $SC_WEAR_TRANSFORM <> 0
; and G56-Reset-Value <> TOWSTD
WidthOfActBitmapsInCharacters = 5
WhichActChannelText=ChannelName,4
; default, show first 4 characters of channel name in lists actual
; tool indicator
; WhichActChannelText=ChannelNumber ;show channel number in lists
; actual tool indicator
; If you need to display more characters of channel name, please
; increase "WidthOfActBitmapsInCharacters".
; For better readability modify bitmaps or use lpaat.bmp,
lpaatd0.bmp,
; lpaatdl0.bmp, lpapt.bmp, lpaptd0.bmp, lpaptdl0.bmp, lpaap.bmp.
; Further you can define a extra column for bitmap exclusive
; display, which will not contain any other data.
; Example:
; [3_ActList]
; ShowActToolCol = 1
; column number where bitmap is displayed
; 1= NoData, 0, Activity
; column 1 is an empty column for display of channel activity;
; column with is 0 + "WidthOfActBitmapsInCharacters";
; column header text is "Activity" or language dependent text;
; File name of the bitmap for the current tool / DNo / DL,
; where D <> 0 and DL <> 0
; ActToolBitmap = paat.bmp
; File name of the bitmap for the current tool / DNo / DL,
; where D = 0 in magazine list and tool list.
; Such cutting edges are not marked in the working offset
; list.
ActToolZeroDBitmap = paatd0.bmp
; File name of the bitmap for the current tool / DNo / DL,
; with current DL = 0.
ActToolZeroDLBitmap = paatdl0.bmp
; as for ActToolBitmap for the programmed tool
ProgToolBitmap = papt.bmp
; as for ActToolZeroDBitmap for the programmed tool
ProgToolZeroDBitmap = paptd0.bmp
; as for ActToolZeroDBitmap for the programmed tool
ProgToolZeroDLBitmap = paptdl0.bmp
; File name of the bitmap for the current magazine location
ActPlaceBitmap = paap.bmp
; show whether the current magazine is free or locked for
; loading/unloading tools.
ShowMagFreeLocked = False
; ShowMagFreeLocked = True
; Name of the bitmap file to show whether the current magazine
; is free for loading/unloading tools.
MagFreeBitmap = magfree.bmp
; Name of the bitmap file to show whether the current magazine
; is locked for loading/unloading tools.
MagLockBitmap = maglock.bmp
; File name of bitmap used in lists to show: G56-reset-value =
; TOWMCS = G56-current-value
G56ResetTOWMCSEqualCurrBitmap = pemcs.bmp
; File name of bitmap used in lists to show: G56-reset-value =
; TOWMCS = G56-current-value
G56ResetTOWWCSEqualCurrBitmap = pewcs.bmp
; File name of bitmap used in lists to show: G56-reset-value =
; TOWSTD = G56-current-value
G56ResetTOWSTDEqualCurrBitmap = pestd.bmp
; File name of bitmap used in lists to show: G56-reset-value =
; TOWMCS <> G56-current-value
G56ResetTOWMCSUnequalCurrBitmap = pumcs.bmp
; File name of bitmap used in lists to show: G56-reset-value =
; TOWMCS <> G56-current-value
G56ResetTOWWCSUnequalCurrBitmap = puwcs.bmp
; File name of bitmap used in lists to show: G56-reset-value =
; TOWSTD <> G56-current-value
G56ResetTOWSTDUnequalCurrBitmap = pustd.bmp
; In order to prevent excessive horizontal scrolling, cursor moves
; automatically to leftmost column in lists during the following
; actions: Magazine lists: softkey "Load" and softkeys "Start"
; and "Abort" in tool loading mode
; tool lists: softkey "New tool".
; CursorMovesLeftmostBySomeActions=True ;default
;CursorMovesLeftmostBySomeActions=False
; In magazine list forms change softkey "Next Mag" to softkey
; "Magazine Selection", to activate vertical softkeys in an
; additional state for magazine selection.
; This helps to prevent users from excessive use of "next mag"
; softkey if a lot of magazines are available.
; You can define shortcut softkeys for up to 5 favorite magazines
; per TOA using section [ShortcutSoftKeysForMagSelect] in
; paramtm.ini and patm_??.ini.
; MagListMagSelectSoftkey=NextMag ; default
;MagListMagSelectSoftkey=SelectMag
; Magazine list: Display buffer initially
DisplayBufferInMagList = InitialVisible
; default behaviour as of version 6.4.1.
; Whenever an other magazine is visited in magazine list:
; automatically show its buffer too.
; The buffer is hidden temporarily after the user has pressed
; the softkey "buffer".
;DisplayBufferInMagList = InitialNotVisible
; old behaviour before version 6.4.1.
; In magazine list: At the beginning no buffer is shown.
; After a magazine which doesn’t have a buffer was visited,
; no buffer is then always shown for other magazines either.
; The buffer is shown temporarily after the user has pressed
; the softkey "buffer".
```

[SoftKeysForMagAndToolList]

```
; Text for the softkeys for magazine lists and tool lists.
; The locale mechanism is used.
; For details of this mechanism, see the section on
; "[GeneralSettingsForMagAndToolList]", entry
; "MagPlaceState_Lang_12345678" or "second "TC_TP2""
1_MagList=M1
2_MagList=M2
3_MagList=M3
1_ToolList=T1
2_ToolList=T2
3_ToolList=T3
1_ActList=A1
2_ActList=A2
3_ActList=A3
```

[FormTitles]

```
; List display titles for the untransformed display of
; data with locale-dependent and transformation-dependent
; mechanisms. For a transformed display the substitute for the
; text codes is taken from the "[TrafoFormTitles]" section.
; If "ReadLanguageIni=1" in "[General]", then the text is
; read from the locale-dependent file
; (e.g. language\patm_gr.ini), see "[FormTitles]" section.
; In each case the text parameters should be set for all
; list displays, for both transformed and
; untransformed display.
; Pay particular attention to the working offset lists: currently
; these lists are only output with transformed data.
; In this case the list display titles from
; "[TrafoFormTitles]" is used.
1_MagList=M1
2_MagList=M2
3_MagList=M3
1_ToolList=T1
2_ToolList=T2
3_ToolList=T3
1_ActList=A1
2_ActList=A2
3_ActList=A3
```

[TrafoFormTitles]

```
; List display titles for untransformed data display
; with locale-dependent and transformation-dependent mechanisms.
; If "ReadLanguageIni=1" in "[General]", then the text is
; read from the locale-dependent file
; see the "[FormTitles]" section.
M1=TM1
M2=TM2
M3=TM3
```

```
T1=TT1
T2=TT2
T3=TT3
A1=TA1
A2=TA2
A3=TA3
```

[SearchOfMagPlaces]

```
; Defines how the location search is to take place.
; Softkey text for 1st location search, locale-dependent
1_SoftkeyText=EL1
; Half location; left, right, top, bottom
1_ToolSizeLRTB=1,1,1,1
; Location type number for location search
1_PlaceTypeNo=1
; Softkey text for 2nd location search, locale-dependent
2_SoftkeyText=EL2
; Half location; left, right, top, bottom
2_ToolSizeLRTB=1,2,1,1
; Location type number for location search
2_PlaceTypeNo=1
; Softkey text for 3rd location search, locale-dependent
3_SoftkeyText=EL3
; Half location; left, right, top, bottom
3_ToolSizeLRTB=2,2,1,1
; Location type number for location search
3_PlaceTypeNo=1
; Softkey text for 3rd location search, locale-dependent
4_SoftkeyText=EL4
; Half location; left, right, top, bottom
4_ToolSizeLRTB=1,2,1,1
; Location type number for location search
4_PlaceTypeNo=2
; Tool OEM Data and Tool Application Data
; Help for UserDataParamIO lines:
; Format of data in magazine list and tool list.
; This format is used only to display data and, in some cases,
; to input data in the HMI, magazine list and tool list.
; In the NC the data format of the OEM data is "float".
; In parameter lines "named parameters" are used. Names are
; separated from the value by ":=". Parameters are separated by ","
; Blanks are allowed on the left and right of the parameter
; name, ":=", value and ",".
; Example for the syntax of lines in the OEM data format:
; "[ToolParams]"
; "UserDataParamIO5=
; Type:=Float, Res:=2,
; Min:=-9999, Max:=9999 ;Comment"
; "UserDataParamIO6=
; Type:=Int, Min:=-99, Max:=99"
; "UserDataParamIO7=
; Type:=Int, Min:=0, Max:=1"
; "5": Number of the OEM data
; "Type:=...": Defines the type of OEM data;
; Default: "Float"
; "Float": Floating point as used in the NC.
; The number of decimal places depends on the MD
; $MM_DISPLAY_RESOLUTION in mmc.ini and on the "res:=..."
; in this parameter line.
; "Int": Integer (range: -999999999 to 999999999)
; "Res:=2": decimal places,
; if "Type:=Float" (floating-point number).
; Range from 0 to 6, default is $MM_DISPLAY_RESOLUTION.
; If "Res:=..." is greater than $MM_DISPLAY_RESOLUTION,
; only the places $MM_DISPLAY_RESOLUTION are displayed.
; Resolution, accuracy: Number of
; "Min:=-9999": For the input: Minimum value
; "Max:=-9999": For the input: Maximum value:
; ";comment": Comment
; Lines UserDataParamName: Locale-specific mechanism
; Lines UserDataParamSize: Locale-specific mechanism
; Lines UserDataParamSizex: Locale-specific mechanism
```

[ToolParams]

```
UserDataParamName1 = TC_TPC1
UserDataParamName2 = TC_TPC2
UserDataParamName3 = TC_TPC3
UserDataParamName4 = TC_TPC4
UserDataParamName5 = TC_TPC5
UserDataParamName6 = TC_TPC6
UserDataParamName7 = TC_TPC7
UserDataParamName8 = TC_TPC8
UserDataParamName9 = TC_TPC9
UserDataParamName10 = TC_TPC10
UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Type:=Float, Res:=2
; UserDataParamIO2 = Type:=Int
UserDataParamSize = TC_TPC_UNIT
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_TPC1_UNIT
; ...
; UserDataParamSize10=TC_TPC10_UNIT
ApplDataParamName1 = TC_TPCS1
ApplDataParamName2 = TC_TPCS2
```

```
ApplDataParamName3 = TC_TPCS3
ApplDataParamName4 = TC_TPCS4
ApplDataParamName5 = TC_TPCS5
ApplDataParamName6 = TC_TPCS6
ApplDataParamName7 = TC_TPCS7
ApplDataParamName8 = TC_TPCS8
ApplDataParamName9 = TC_TPCS9
ApplDataParamName10 = TC_TPCS10
ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Type:=Float, Res:=2
; ApplDataParamIO2 = Type:=Int
ApplDataParamSize = TC_TPCS_UNIT
ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_TPCS1_UNIT
; ...
; ApplDataParamSize10=TC_TPCS10_UNIT
; Example for the syntax of lines in the OEM data format:
; "[ToolEdgeParams]"
; "UserDataParamIO5=
; Type:=Float, Res:=2, Min:=-9999, Max:=9999 "Comment"
; "UserDataParamIO6=
; Type:=Int, Min:=-99, Max:=99"
; "UserDataParamIO7=
; Type:=Int, Min:=0, Max:=1"
; "5": Number of the OEM data
; "Type:=...": Defines the type of OEM data
; Default: "Floatn"
; "Float": Floating point as used in the NC. The number of
; decimal places depends on $MM_DISPLAY_RESOLUTION
; in mmc.ini and on the "res." in this parameter line.
; "Int": Integer (range: -999999999 to 999999999)
; "Res:=2": Resolution, accuracy: Number of
; decimal places if "Type:=Float" (floating-point number).
; Range from 0 to 6, default is $MM_DISPLAY_RESOLUTION.
; If "Res:=..." is greater than $MM_DISPLAY_RESOLUTION,
; only the places $MM_DISPLAY_RESOLUTION are displayed.
; "Min:=-9999": For the input: Minimum value
; "Max:=-9999": For the input: Maximum value:
; ";comment": Comment;
; Lines EdgeParamName...:
; Example: ; EdgeParamNameLLen1=TC_DP3
; Defines the text under which the cutting parameters and
; additive offset parameters in the tool detail screens
; are displayed.
; The sequence of the EdgeParamName... lines corresponds to
; the sequence in the screens.
; The sequence in the screens cannot be influenced by changing
; the sequence in paramtm.ini.
; For the values of EdgeParamName... we used the character string
; "TC_DP..." to indicate how the texts relate to the
; associated NCK variables. However, other
; character strings can be used here because this text
; reference only defines a parameter heading and does not
; specify which data is read from the NCK.
; The value of EdgeParamName... is copied directly into the screen
; if [General] ReadLanguageIni=0 and in the tool
; detail screens the untransformed display is active.
; If [General] ReadLanguageIni=1 and in the tool
; detail screens the untransformed display is active, the
; value of EdgeParamName... is used as the access code to
; read the text from the locale-dependent INI file
; (language\patm_xx.ini, section [ToolEdgeParams], access code
; "TC_DP3" in this example).
; If in the tool detail screens the transformed display
; is active, the value of EdgeParamName... as access code
; is used for the [ToolEdgeParamsTrafoTextReplace] section in
paramtm.ini
; used. ; The value found there is then used as text or
; access code (depending on [General] ReadLanguageIni)
; in order to determine the output text for the tool detail screens.
; If an entry is not found in the INI files, the
; text is taken from the resource file language\pa_xx.dll.
; Lines UserDataParamName: Locale-specific mechanism
; Lines UserDataParamSize: Locale-specific mechanism
; Lines UserDataParamSizex: Locale-specific mechanism
; Lines EdgeParamName...: untransformed or
; transformed mechanism and locale-dependent mechanism.
```

[ToolEdgeParams]

```
; Length 1
EdgeParamNameLLen1 = TC_DP3
; Length 2
EdgeParamNameLLen2 = TC_DP4
; Length 3
EdgeParamNameLLen3 = TC_DP5
; Radius l1
EdgeParamNameRLen1 = TC_DP8
; Radius l2
EdgeParamNameRLen2 = TC_DP9
; Radius r1
EdgeParamNameRRad1 = TC_DP6
; Radius r2
EdgeParamNameRRad2 = TC_DP7
; Angle1
```

```
EdgeParamNameAng1 = TC_DP10
; Angle2
EdgeParamNameAng2 = TC_DP11
UserDataParamName1 = TC_DPC1
UserDataParamName2 = TC_DPC2
UserDataParamName3 = TC_DPC3
UserDataParamName4 = TC_DPC4
UserDataParamName5 = TC_DPC5
UserDataParamName6 = TC_DPC6
UserDataParamName7 = TC_DPC7
UserDataParamName8 = TC_DPC8
UserDataParamName9 = TC_DPC9
UserDataParamName10 = TC_DPC10
UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Type:=Float, Res:=2
; UserDataParamIO2 = Type:=Int
UserDataParamSize = TC_DPC_UNIT
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_DPC1_UNIT
; ...
; UserDataParamSize10=TC_DPC10_UNIT
ApplDataParamName1 = TC_DPCS1
ApplDataParamName2 = TC_DPCS2
ApplDataParamName3 = TC_DPCS3
ApplDataParamName4 = TC_DPCS4
ApplDataParamName5 = TC_DPCS5
ApplDataParamName6 = TC_DPCS6
ApplDataParamName7 = TC_DPCS7
ApplDataParamName8 = TC_DPCS8
ApplDataParamName9 = TC_DPCS9
ApplDataParamName10 = TC_DPCS10
ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Type:=Float, Res:=2
; ApplDataParamIO2 = Type:=Int
ApplDataParamSize = TC_DPCS_UNIT
ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_DPCS1_UNIT
; ...
; ApplDataParamSize10=TC_DPCS10_UNIT
```

[ToolEdgeParamsTrafoTextReplace]

```
; Transformed length 1
TC_DP3 = TTC_DP3
; Transformed length 2
TC_DP4 = TTC_DP4
; Transformed length 3
TC_DP5 = TTC_DP5
```

```
; Transformed radius l1
TC_DP8 = TTC_DP8
; Transformed radius l2
TC_DP9 = TTC_DP9
; Transformed radius r1
TC_DP6 = TTC_DP6
; Transformed radius r2
TC_DP7 = TTC_DP7
; Transformed angle1
TC_DP10 = TTC_DP10
; Transformed angle2
TC_DP11 = TTC_DP11
```

[CuttEdgeSupervisionOEM]

```
; [CuttEdgeSupervisionOEM] cutting edge monitoring OEM data
; and application data
; [MagazineOEM] magazine OEM data and application data
; [MagazineLocOEM] magazine location OEM data and application data
; Help for the lines "UserDataParamIO":
; Data format for the data display in magazine list and
; tool list. ; This format is only used for the display and
; in a number of instances as well for entering data.
; "Name parameters" are used in parameter lines.
; The names are separated from the parameter value by ": = ".
; The parameters themselves are separated one each other by ",";
; blanks are permitted to the left and to the right of parameter
names, ":=",
; parameter value and ",".
; Example for the syntax of lines in the OEM data format:
; "[CuttEdgeSupervisionOEM]"
; "UserDataParamIO5=Min:=-9999, Max:=9999 ;comment"
; "5": Number of the OEM data
; "Min:=-9999": For the input: Minimum value
; "Max:=-9999": For the input: Maximum value:
; ";comment": Comment
; Lines UserDataParamName: Locale-specific mechanism
; Lines UserDataParamSize: Locale-specific mechanism
; Lines UserDataParamSizex: Locale-specific mechanism
```

[CuttEdgeSupervisionOEM]

```
UserDataParamName1 = TC_MOPC1
UserDataParamName2 = TC_MOPC2
UserDataParamName3 = TC_MOPC3
UserDataParamName4 = TC_MOPC4
UserDataParamName5 = TC_MOPC5
UserDataParamName6 = TC_MOPC6
UserDataParamName7 = TC_MOPC7
```

```
UserDataParamName8 = TC_MOPC8
UserDataParamName9 = TC_MOPC9
UserDataParamName10 = TC_MOPC10
UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Min:=-4, Max:=6
; UserDataParamIO2 = Min:=-10, Max:=122
UserDataParamSize = TC_MOPC_UNIT
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_MOPC1_UNIT
; ...
; UserDataParamSize10=TC_MOPC10_UNIT
ApplDataParamName1 = TC_MOPCS1
ApplDataParamName2 = TC_MOPCS2
ApplDataParamName3 = TC_MOPCS3
ApplDataParamName4 = TC_MOPCS4
ApplDataParamName5 = TC_MOPCS5
ApplDataParamName6 = TC_MOPCS6
ApplDataParamName7 = TC_MOPCS7
ApplDataParamName8 = TC_MOPCS8
ApplDataParamName9 = TC_MOPCS9
ApplDataParamName10 = TC_MOPCS10
ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Min:=-4, Max:=6
; ApplDataParamIO2 = Min:=-10, Max:=122
ApplDataParamSize = TC_MOPCS_UNIT
ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_MOPCS1_UNIT
; ...
; ApplDataParamSize10=TC_MOPCS10_UNIT
```

[MagazineOEM]

```
UserDataParamName1 = TC_MAPC1
UserDataParamName2 = TC_MAPC2
UserDataParamName3 = TC_MAPC3
UserDataParamName4 = TC_MAPC4
UserDataParamName5 = TC_MAPC5
UserDataParamName6 = TC_MAPC6
UserDataParamName7 = TC_MAPC7
UserDataParamName8 = TC_MAPC8
UserDataParamName9 = TC_MAPC9
UserDataParamName10 = TC_MAPC10
UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Min:=-22, Max:=24
; UserDataParamIO2 = Min:=-10, Max:=162
UserDataParamSize = TC_MAPC_UNIT
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_MAPC1_UNIT
; ...
; UserDataParamSize10=TC_MAPC10_UNIT
ApplDataParamName1 = TC_MAPCS1
ApplDataParamName2 = TC_MAPCS2
ApplDataParamName3 = TC_MAPCS3
ApplDataParamName4 = TC_MAPCS4
ApplDataParamName5 = TC_MAPCS5
ApplDataParamName6 = TC_MAPCS6
ApplDataParamName7 = TC_MAPCS7
ApplDataParamName8 = TC_MAPCS8
ApplDataParamName9 = TC_MAPCS9
ApplDataParamName10 = TC_MAPCS10
ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Min:=-22, Max:=24
; ApplDataParamIO2 = Min:=-10, Max:=162
ApplDataParamSize = TC_MAPCS_UNIT
ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_MAPCS1_UNIT
; ...
; ApplDataParamSize10=TC_MAPCS10_UNIT
```

[MagazineLocOEM]

```
UserDataParamName1 = TC_MPPC1
UserDataParamName2 = TC_MPPC2
UserDataParamName3 = TC_MPPC3
UserDataParamName4 = TC_MPPC4
UserDataParamName5 = TC_MPPC5
UserDataParamName6 = TC_MPPC6
UserDataParamName7 = TC_MPPC7
UserDataParamName8 = TC_MPPC8
UserDataParamName9 = TC_MPPC9
UserDataParamName10 = TC_MPPC10
UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Min:=-42, Max:=62
; UserDataParamIO2 = Min:=-210, Max:=712
UserDataParamSize = TC_MPPC_UNIT
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_MPPC1_UNIT
; ...
; UserDataParamSize10=TC_MPPC10_UNIT
ApplDataParamName1 = TC_MPPCS1
ApplDataParamName2 = TC_MPPCS2
ApplDataParamName3 = TC_MPPCS3
ApplDataParamName4 = TC_MPPCS4
ApplDataParamName5 = TC_MPPCS5
ApplDataParamName6 = TC_MPPCS6
ApplDataParamName7 = TC_MPPCS7
```

```
ApplDataParamName8 = TC_MPPCS8
ApplDataParamName9 = TC_MPPCS9
ApplDataParamName10 = TC_MPPCS10
ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Min:=-42, Max:=62
; ApplDataParamIO2 = Min:=-210, Max:=712
ApplDataParamSize = TC_MPPCS_UNIT
ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_MPPCS1_UNIT
; ...
; ApplDataParamSize10=TC_MPPCS10_UNIT
```

Parameterization of the individual magazine, tool and working offset lists

```
; From changes in the following sections it is possible to determine
; which data is to be displayed in the individual magazine, tool and
; working offset lists.
; [1_MagList], [2_MagList], [3_MagList],
; [1_ToolList], [2_ToolList], [3_ToolList],
; [1_ActList], [2_ActList], [3_ActList].
; In these sections you can determine the number of
; columns that are not moved (i.e. always visible) during
; horizontal navigation ("NrOfFixedColumns=") and the
; individual columns "1=...", "2=...", etc.).
; The column number (number in front of "=") can take a value
between
; 1 and 1000. The maximum number of columns in a
; list is around 90, although with 90 columns in a
; list the display speed will slow down and
; the user will have to scroll horizontally in order to be able to
; view all the columns, so as a general rule this limit is
; not reached.
; Gaps are allowed between the numbers in the sequence of
; column numbers.
; To disable a column predefined in mmc2\paramtm.ini,
; in the user\paramtm.ini you can
; add the corresponding entry with the value "<Empty>".
; Specify "...=<EndOfList>" to define the end of
; the list. This increases the speed when reading
; the INI files after starting tool management.
; Example for the syntax of a column-definition line:
; "2=TC_TP2,11,TC_TP2 ; WzIdent "2": Number of the entry
; for first "TC_TP2": determines which NC data in the list of
columns
; will be displayed. The character string TC_TP2 corresponds to an
; NCK variable, see programming manual.
; The specified character strings are described in paramtm.txt.
; New in Version P5:
```

```
; If "MultiLine=SINGLE" appears in a list definition section of a
; magazine or tool list, then the cutting edge number
; can be specified by adding "@Ee", where "e" is the
; cutting edge number (range from 1 to the maximum number of
; cutting edges per tool) for all cutting edge data.
; This applies to the following data:
; Cutting edge data TC_DPp@Ee
; Cutting edge monitoring data TC_MOPp@Ee
; OEM cutting edge data TC_DPCp@Ee
; Freely assignable D No. TC_DPCE@Ee
; Additive offset TC_SCPz@Ee
; Setup compensation TC_ECPz@Ee
; If "@Ee" is not specified in these columns, then the data for
; cutting edge 1 is used. This method of proceeding is
; compatible with the earlier versions of P5. To avoid confusion,
; for these cutting edges you should specify the cutting edge
; number in the header text for the corresponding column.
; "@Ee" cannot be specified in magazine and tool lists
; for which "Multitime=MULTI" or with working offset lists.
; These values automatically display the data for the current
; cutting edges.
; "11": Approximate column width in characters, based on
; "[GeneralSettingsForMagAndToolList]", entries
; "ColumnWidthTwipsPer-AlphaCharacter" and
; "ColumnWidthTwipsPerNumericCharacter"
; second "TC_TP2": Column header text or code for
; text. If in the "[General]" section the entry for
; "ReadLanguageIni"="1", then the system searches for the column
header text
; in the files mmc2\language\patm_gr.ini,
; user\language\patm_gr.ini etc. in the section
; "[ListColumnHeaderText]", entry "TC_TP2" (in this example).
; The character string "gr" in "patm_gr.ini" depends on the
; particular language (see mmc.ini, "[LANGUAGE]", entry
; "Language=...").
; If the entry in Language\patm_gr.ini is found,
; then it is used as the column header text.
; If no entry is found or if the text "...=<Empty>"
; then the value from the paramtm.ini file
; is used as the column header text.
; If transformed data is displayed, then the column
; header text or its access code is also "transformed",
; in which the corresponding assignment of the section
; [ListColumnHeaderTrafoTextReplace] is used for substitution.
; (Therefore both the transformed/untransformed mechanism
; and the locale mechanism are used.)
; "WzIdent": ";" introduces a comment; at the end of a
; parameter line you can also introduce a comment with "//".
```

[1_MagList]

```
MultiLine=SINGLE
NrOfFixedColumns=1
1= ToolInPlace, 3, ToolInPlace
2= TC_MPP4_1, 1, TC_MPP4_1
3= TC_MPP4_2, 1, TC_MPP4_2
4= TC_MPP4_3, 1, TC_MPP4_3
5= TC_MPP4_4, 1, TC_MPP4_4
6= TC_MPP4_5, 1, TC_MPP4_5
7= TC_MPP4_6, 1, TC_MPP4_6
8= TC_MPP4_7, 1, TC_MPP4_7
9= TC_MPP4_8, 1, TC_MPP4_8
10= TC_TP2, 11, TC_TP2
11= TC_TP1, 5, TC_TP1
12= TC_MPP6, 5, TC_MPP6
13= TC_TP3, 1, TC_TP3
14= TC_TP4, 1, TC_TP4
15= TC_TP5, 1, TC_TP5
16= TC_TP6, 1, TC_TP6
17= TC_TP7, 4, TC_TP7
18= TC_TP8_1, 1, TC_TP8_1
19= TC_TP8_2, 1, TC_TP8_2
20= TC_TP8_3, 1, TC_TP8_3
21= TC_TP8_4, 1, TC_TP8_4
22= TC_TP8_5, 1, TC_TP8_5
23= TC_TP8_6, 1, TC_TP8_6
24= TC_TP8_7, 1, TC_TP8_7
25= TC_TP8_8, 1, TC_TP8_8
26= NoData, 1, <automatic extend last column>
27= <EndOfList>
```

[2_MagList]

```
MultiLine=MULTI
NrOfFixedColumns=1
1= ToolInPlace, 3, ToolInPlace
2= TC_TP2, 11, TC_TP2
3= TC_TP1, 5, TC_TP1
4= TC_MPP6, 5, TC_MPP6
5= TC_TP7, 4, TC_TP7
6= CuttEdgeNo, 1, CuttEdgeNo
7= TC_DP1, 4, TC_DP1
8= TC_DP3, 11, TC_DP3
9= TC_DP6, 11, TC_DP6
10= TC_TP3, 1, TC_TP3
11= TC_TP4, 1, TC_TP4
```

```
12= TC_TP5, 1, TC_TP5
13= TC_TP6, 1, TC_TP6
14= TC_MPP2, 4, TC_MPP2
15= TC_ADPT1, 11, TC_ADAPT1
16= TC_ADPT2, 11, TC_ADAPT2
17= TC_ADPT3, 11, TC_ADAPT3
18= TC_ADPT4, 4, TC_ADAPT4
19= <EndOfList>
```

[3_MagList]

```
NrOfFixedColumns=1
1= ToolInPlace, 3, ToolInPlace
2= TC_TP2, 11, TC_TP2
3= TC_TP1, 5, TC_MPP6
5= TC_TP9, 1, TC_TP9
6= TC_MOP1, 7, TC_MOP1
7= TC_MOP2, 7, TC_MOP2
8= TC_MOP3, 7, TC_MOP3
9= TC_MOP4, 7, TC_MOP4
10= TC_MPP3, 1, TC_MPP3
11= TC_MPP5, 2, TC_MPP5
12= NoData, 1, <automatic extend last column>
13= <EndOfList>
```

[1_ToolList]

```
NrOfFixedColumns=1
1= NO, 4, NO
2= MagNo, 4, MagNo
3= ToolInPlace, 3, ToolInPlace
4= TC_TP2, 11, TC_TP2
5= TC_TP1, 5, TC_TP1
6= TC_MPP6, 5, TC_MPP6
7= TC_TP3, 1, TC_TP3
8= TC_TP4, 1, TC_TP4
9= TC_TP5, 1, TC_TP5
10= TC_TP6, 1, TC_TP6
11= TC_TP8_1, 1, TC_TP8_1
12= TC_TP8_2, 1, TC_TP8_2
13= TC_TP8_3, 1 TC_TP8_3
14= TC_TP8_4, 1, TC_TP8_4
15= TC_TP8_5, 1, TC_TP8_5
16= TC_TP8_6, 1, TC_TP8_6
17= TC_TP8_7, 1, TC_TP8_7
18= TC_TP8_8, 1, TC_TP8_8
19= TC_TP7, 4, TC_TP7
20= NoData, 1, <automatic extend last column>
```

```
21= <EndOfList>
```

[2_ToolList]

```
MultiLine=MULTI
NrOfFixedColumns=1
1= NO, 4, NO
2= MagNo, 4, MagNo
3= ToolInPlace, 3, ToolInPlace
4= TC_TP2, 11, TC_TP2
5= TC_TP1, 5, TC_TP1
6= TC_MPP6, 5, TC_MPP6
7= CuttEdgeNo, 1, CuttEdgeNo
8= TC_DP1, 4, TC_DP1
9= TC_DP3, 11, TC_DP3
10= TC_DP4, 11, TC_DP4
11= TC_DP5, 11, TC_DP5
12= TC_DP6, 11, TC_DP6
13= TC_MPP2, 4, TC_MPP2
14= <EndOfList>
```

[3_ToolList]

```
NrOfFixedColumns=3
1= NO, 4, NO
2= MagNo, 4, MagNo
3= ToolInPlace, 3, ToolInPlace
4= TC_TP2, 11, TC_TP2
5= TC_TP1, 5, TC_TP1
6= TC_MPP6, 5, TC_MPP6
7= TC_TP3, 1, TC_TP3
8= TC_TP4, 1, TC_TP4
9= TC_TP5, 1, TC_TP5
10= TC_TP6, 1, TC_TP6
11= TC_DP3@E1, 11, e1TC_DP3
12= TC_DP3@E2, 11, e2TC_DP3
13= TC_DP3@E3, 11, e3TC_DP3
14= TC_TP9, 1, TC_TP9
15= TC_MOP1, 7, TC_MOP1
16= TC_MOP2, 7, TC_MOP2
17= TC_MOP3, 7, TC_MOP3
18= TC_MOP4, 7, TC_MOP4
19= <EndOfList>
```

[1_ActList]

```
MultiLine=SINGLE
NrOfFixedColumns=1
NumLinesPerReq = 11
```

```
1= NO, 4, NO
2= TC_TP2, 11, TC_TP2
3= TC_TP1, 5, TC_TP1
4= TC_MPP6, 5, TC_MPP6
5= CuttEdgeNo, 1, CuttEdgeNo
6= TC_DPCE, 6, TC_DPCE
7= MagNo, 4, MagNo
8= ToolInPlace, 3, ToolInPlace
9= TC_MPP2, 3, TC_MPP2
10= TC_MPP5, 4, TC_MPP5
11= TC_DP1, 11, TC_DP3
13= TC_DP4, 11, TC_DP4
14= TC_SCP13, 9, TC_SCP13
15= TC_SCP14, 9, TC_SCP14
16= TC_SCP23, 9, TC_SCP23
17= TC_SCP24, 9, TC_SCP24
18= TC_ADPT1, 11, TC_ADAPT1
19= TC_ADPT2, 11, TC_ADAPT2
20= TC_ADPT3, 11, TC_ADAPT3
21= TC_ADPT4, 4, TC_ADAPT4
22= TC_TP8_1, 1, TC_TP8_1
23= TC_TP8_2, 1, TC_TP8_2
24= TC_TP8_3, 1, TC_TP8_3
25= TC_TP8_4, 1, TC_TP8_4
26= TC_TP8_5, 1, TC_TP8_5
27= TC_TP8_8, 1, TC_TP8_8
28= <EndOfList>
```

[2_ActList]

```
MultiLine=MULTI
NrOfFixedColumns=0
NumLinesPerReq = 6
1= NO, 4, NO
2= TC_TP2, 11, TC_TP2
3= TC_TP1, 5, TC_TP1
4= TC_MPP6, 5, TC_MPP6
5= CuttEdgeNo, 1, CuttEdgeNo
6= TC_DPCE, 6, TC_DPCE
7= MagNo, 4, MagNo
8= ToolInPlace, 3, ToolInPlace
9= DLNO, 3, DLNO
10= TC_SCP3, 9, TC_SCP3
11= TC_SCP4, 9, TC_SCP4
12= <Empty>
13= <Empty>
14= TC_DP1, 4, TC_DP1
15= TC_DP2, 11, TC_DP2
```

```
16= TC_DP3, 11, TC_DP3
17= TC_DP4, 11, TC_DP4
18= <Empty>
19= <Empty>
20= TC_ADPT1, 11, TC_ADAPT1
21= TC_ADPT2, 11, TC_ADAPT2
22= TC_ADPT3, 11, TC_ADAPT3
23= TC_ADPT4, 4, TC_ADAPT4
24= <EndOfList>
```

[3_ActList]

```
NrOfFixedColumns=3
NumLinesPerReq = 11
1= NO, 4, NO
2= MagNo, 4, MagNo
3= ToolInPlace, 3, ToolInPlace
4= TC_TP2, 11, TC_TP2
5= TC_TP1, 5, TC_TP1
6= TC_DPCE, 6, TC_DPCE
7= TC_TP3, 1, TC_TP3
8= TC_TP4, 1, TC_TP4
9= TC_TP5, 1, TC_TP5
10= TC_TP6, 1, TC_TP6
11= TC_TP9, 1, TC_TP9
12= TC_MOP1, 7, TC_MOP1
13= TC_MOP2, 7, TC_MOP2
14= TC_MOP3, 7, TC_MOP3
15= TC_MOP4, 7, TC_MOP4
16= TC_MOP5, 7, TC_MOP5
17= TC_MOP6, 7, TC_MOP6
18= TC_MOP11, 7, TC_MOP11
19= TC_MOP13, 7, TC_MOP13
20= TC_MOP15, 7, TC_MOP15
21= <EndOfList>
```

[ListColumnHeaderTrafoTextReplace]

```
; Example:
; TC_DP3 = TTC_DP3
; In the transformed display of lists, this determines the
substitute text
; for the cutting parameters and additive offset parameters
; in the column headers.
; The substitute code for the transformed presentation is searched
for
; in the code "TC_DP3".
; If an entry is missing in this section, then in the
; column header for the corresponding column the
; message "missed trafo text" is
; displayed.
TC_DP3 = TTC_DP3
e1TC_DP3 = Te1TC_DP3
e2TC_DP3 = Te2TC_DP3
e3TC_DP3 = Te3TC_DP3
TC_DP4 = TTC_DP4
TC_DP5 = TTC_DP5
TC_DP6 = TTC_DP6
TC_DP7 = TTC_DP7
TC_DP8 = TTC_DP8
TC_DP9 = TTC_DP9
TC_DP10 = TTC_DP10
TC_DP11 = TTC_DP11
TC_DP12= TTC_DP12
TC_DP13 = TTC_DP13
TC_DP14 = TTC_DP14
TC_DP15 = TTC_DP15
TC_DP16 = TTC_DP16
TC_DP17 = TTC_DP17
TC_DP18 = TTC_DP18
TC_DP19 = TTC_DP19
TC_DP20 = TTC_DP20
TC_DP21 = TTC_DP21
TC_DP22 = TTC_DP22
TC_DP23 = TTC_DP23
TC_DP24 = TTC_DP24
TC_DP25 = TTC_DP25
TC_DPH = TTC_DPH
TC_DPV = TTC_DPV
TC_DPV3 = TTC_DPV3
TC_DPV4 = TTC_DPV4
TC_DPV5 = TTC_DPV5
TC_SCP3 = TTC_SCP3
TC_SCP4 = TTC_SCP4
TC_SCP5 = TTC_SCP5
TC_SCP6 = TTC_SCP6
TC_SCP7 = TTC_SCP7
TC_SCP8 = TTC_SCP8
TC_SCP9 = TTC_SCP9
TC_SCP10 = TTC_SCP10
TC_SCP11 = TTC_SCP11
TC_SCP13 = TTC_SCP13
TC_SCP14 = TTC_SCP14
TC_SCP15 = TTC_SCP15
TC_SCP23 = TTC_SCP23
TC_SCP24 = TTC_SCP24
```

```
TC_SCP25 = TTC_SCP25
TC_SCP33 = TTC_SCP33
TC_SCP34 = TTC_SCP34
TC_SCP35 = TTC_SCP35
e1TC_SCP13 = Te1TC_SCP13
e1TC_SCP14 = Te1TC_SCP14
e1TC_SCP15 = Te1TC_SCP15
e1TC_SCP23 = Te1TC_SCP23
e1TC_SCP24 = Te1TC_SCP24
e1TC_SCP25 = Te1TC_SCP25
e1TC_SCP33 = Te1TC_SCP33
e1TC_SCP34 = Te1TC_SCP34
e1TC_SCP35 = Te1TC_SCP35
e2TC_SCP13 = Te2TC_SCP13
e2TC_SCP14 = Te2TC_SCP14
e2TC_SCP15 = Te2TC_SCP15
e2TC_SCP23 = Te2TC_SCP23
e2TC_SCP24 = Te2TC_SCP24
e2TC_SCP25 = Te2TC_SCP25
e2TC_SCP33 = Te2TC_SCP33
e2TC_SCP34 = Te2TC_SCP34
e2TC_SCP35 = Te2TC_SCP35
e3TC_SCP13 = Te3TC_SCP13
e3TC_SCP14 = Te3TC_SCP14
e3TC_SCP15 = Te3TC_SCP15
e3TC_SCP23 = Te3TC_SCP23
e3TC_SCP24 = Te3TC_SCP24
e3TC_SCP25 = Te3TC_SCP25
e3TC_SCP33 = Te3TC_SCP33
e3TC_SCP34 = Te3TC_SCP34
e3TC_SCP35 = Te3TC_SCP35
TC_ECP3 = TTC_ECP3
TC_ECP4 = TTC_ECP4
TC_ECP5 = TTC_ECP5
TC_ECP6 = TTC_ECP6
TC_ECP7 = TTC_ECP7
TC_ECP8 = TTC_ECP8
TC_ECP9 = TTC_ECP9
TC_ECP10 = TTC_ECP10
TC_ECP11 = TTC_ECP11
TC_ECP13 = TTC_ECP13
TC_ECP14 = TTC_ECP14
TC_ECP15 = TTC_ECP15
TC_ECP23 = TTC_ECP23
TC_ECP24 = TTC_ECP24
TC_ECP25 = TTC_ECP25
TC_ECP33 TTC_ECP33
```

```
TC_ECP34 = TTC_ECP34
TC_ECP35 = TTC_ECP35
e1TC_ECP13 = Te1TC_ECP13
e1TC_ECP14 = Te1TC_ECP14
e1TC_ECP15 = Te1TC_ECP15
e1TC_ECP23 = Te1TC_ECP23
e1TC_ECP24 = Te1TC_ECP24
e1TC_ECP25 = Te1TC_ECP25
e1TC_ECP33 = Te1TC_ECP33
e1TC_ECP34 = Te1TC_ECP34
e1TC_ECP35 = Te1TC_ECP35
e2TC_ECP13 = Te2TC_ECP13
e2TC_ECP14 = Te2TC_ECP14
e2TC_ECP15 = Te2TC_ECP15
e2TC_ECP23 = Te2TC_ECP23
e2TC_ECP24 = Te2TC_ECP24
e2TC_ECP25 = Te2TC_ECP25
e2TC_ECP33 = Te2TC_ECP33
e2TC_ECP34 = Te2TC_ECP34
e2TC_ECP35 = Te2TC_ECP35
e3TC_ECP13 = Te3TC_ECP13
e3TC_ECP14 = Te3TC_ECP14
e3TC_ECP15 = Te3TC_ECP15
e3TC_ECP23 = Te3TC_ECP23
e3TC_ECP24 = Te3TC_ECP24
e3TC_ECP25 = Te3TC_ECP25
e3TC_ECP33 = Te3TC_ECP33
e3TC_ECP34 = Te3TC_ECP34
e3TC_ECP35 = Te3TC_ECP35
```

[BatchTools]

```
; Control of the job functions for the tools:
; load, unload or reactivate a number of tools
; Note: The tool filters only work if
; bit 4 (from 0 to ...) is set in the ToolManagementMask.
; Max. 6 filters can be specified.
; The following can be specified for each filter:
; Softkey text, list header, search criteria, selection of the
; type of results list and additional data.
; The ...user\paramini.out file contains error messages for
; errors found on reading the parameters.
; Search criteria:
; Permissible values in "FindCondition":
; A maximum of 8 entries are permissible, separated by ",".
; They are ANDed. In the part conditions
; no data may occur more than once. Each part condition consists
; of three parts:
```

```
; 1. Date for which the condition applies
; 2. Condition
; 3. Comparison value
; The following data can be a filter criterion:
; Tool data:
TC-TP1 Duplo number
TC-TP2 Tool identifier
TC-TP3 Tool size in half locations left
TC-TP4 Tool size in half locations right
TC-TP5 Tool size in half locations top
TC-TP6 Tool size in half locations bottom
TC-TP7 Tool location type
TC-TP8 Tool status
TC-TP9 Monitoring type
TC-TP10 Replacement tool search
TC-TP11 Tool information/replacement tool sequence
A_TOOLMN Magazine number
A_TOOLMLN Magazine location number
P_TOOLND Number of cutting edges
; Tool OEM data: "TC_TPC1" to "TC_TPC10"
; The tool OEM data must be activated on the NC
; and the numbers permissible.
; Tool cutting edge parameters:
"TC_DP1" to "TC_DP25", "TC_DPH", "TC_DPV", "TC_DPV3", "TC_DPV4",
"TC_DPV5" (the NCK setting applies instead of "25")
; Tool cutting edge OEM data: "TC_DPC1" to "TC_DP10"
; The tool cutting edge OEM data must be activated on the NC
; and the numbers must be permissible on the NC.
; Tool cutting edge monitoring parameters:
TC_MOP1 Prewarning limit for tool life
TC_MOP2 Actual value for tool life
TC_MOP3 Prewarning limit for workpiece count
TC_MOP4 Actual value for workpiece count
TC_MOP5 Prewarning limit for wear
TC_MOP6 Remaining wear
TC_MOP11 Setpoint for tool life
TC_MOP13 Setpoint for tool workpiece count
TC_MOP15 Setpoint for wear
; Tool cutting edge monitoring OEM data "TC_MOPC1" to
; "TC_MOPC10" tool cutting edge monitoring OEM data must be
; activated on the NC and the numbers must be permissible
; on the NCK. If the NCK version >= 430000: User data
; Tool user data:
; "TC_TPCS1" to "TC_TPCS10"
; Tool user data must be activated on the NC and
; the numbers must be permissible on the NC.
; Tool cutting edge user data:
```

```
; "TC_DPCS1" to "TC_DPCS10"
; The tool cutting edge user data must be activated on the NC
; and the numbers must be permissible on the NC.
; Cutting edge monitoring user data:
; "TC_MOPCS1" to "TC_MOPCS10"
; The cutting edge monitoring user data must be activated on the NC
; and the numbers must be permissible on the NC.
; Condition:
"==" equal
">" less than
">" greater than
"<=" less than or equal to
">=" greater than or equal to
; && Bit-wise AND, only permissible for operands of type
; WORD and DOUBLEWORD
; "==" is the only relational operator allowed for
; string operands
; Comparison value: String for TC_TP2 (tool data,
; tool identifier),
; max. 32 characters, no spaces before or after
; 0 ... 65535 for all other TC_TP data
; Double for all other data
; Max. one column with additional data can appear for
; each filter.
; _FindResultAddColumnBtss:
; Additional data, OPI item acc. to OPI documentation
(mmc2\btss_gr.hlp).
; Example 1: "/Tool/User/data[u#TOA#,c2,#TNO#](|"!d%.#RES#lf")"
; Tool OEM parameter 2, floating-point notation
; Standard number of decimal places
; Example 2: "/Tool/User/data[u#TOA#,c3,#TNO#](|"!l%ld")"
; Tool OEM parameter 3, integer notation
; Example 3:
"/Tool/MagazineDescription/userData[u#TOA#,c#MAG#,1](|)"
; Magazine OEM parameter 1
; The following placeholders are permissible: #TOA#, #TNO#, #MAG#,
#RES#.
; #TOA# TOA number (of the current channel)
; #TNO# Internal T number (of the tool found)
; #MAG# Magazine number (of the tool found)
; #RES# Standard value for the number of decimal places
; Placeholders are substituted by the data for the current tool or
; by general settings.
; Max. 1 OPI item is permitted.
; "(|)" is entered in front of the result data to generate the data
; separation character "|".
; OPI multiple variable accesses are generated internally from the
OPI item.
; The OPI item must be enclosed by " ",
; especially if it contains formatting information
; enclosed by " ".
; The user setting the parameters is responsible for the
; correct syntax. The syntax is not checked by the OPI.
; General settings for all filters:
; This entry applies for HMI_ADV prior to software Version 6.3.
; With Version 6.3 and higher it is ignored.
; Colors for the Results list:
; A hex value consisting of 8 characters is assigned to
; each color. The hex value has the following syntax:
; SSBBGGRR where SS=System, BB=Blue, GG=Green, RR=Red
; The colors have to be specified for the following list elements:
; Non-selected text
; Non-selected background
; Cursor-selected text
; Cursor-selected background
; Job-selection and cursor-selected text
; Job-selection and cursor-selected background
; Job-selection and cursor-selected text
; Job-selection and cursor-selected background
; WinTxt, WinBa, HighLTxt, HighLiBa, TiBaTxt green,
; TiBaTxt bluegreen
; ResultColors=80000008,80000005,8000000E,8000000D,80000009,
; 0000FF00,80000009,00FF8000
;experimental, LiteBlue for batch selected
; WinTxt, WinBa, HighLTxt, HighLiBa, WinTxt ,LiteBlue,
; HighLTxt, HighLiBa
; ResultColors=80000008,80000005,8000000E,8000000D,80000008,
; 00FFFF00,8000000E,8000000D
;experimental, LiteGreen for batch selected
;WinTxt, WinBa, HighLTxt,HighLiBa,WinTxt ,Litegree,HighLTxt,HighLiBa
ResultColors=80000008,80000005,8000000E,8000000D,80000008,
; 000FF000,8000000E,8000000D
; WinTxt, WinBa,
HighLTxt,HighLiBa,HighLTxt,HighLiBa,HighLTxt,HighLiBa
ResultColors=80000008,80000005,8000000E,8000000D,8000000E,
; 8000000D,8000000E,8000000D
; The user can replace the names of the bitmaps or the bitmaps
; themselves with custom bitmaps. Custom bitmaps
; are stored in the "user" directory.
BatchFilterElBUnTUnBitmap = pbfbutu.bmp
BatchFilterElBUnTSeBitmap = pbfbuts.bmp
BatchFilterElBSeTUnBitmap = pbfbstu.bmp
BatchFilterElBSeTSeBitmap = pbfbsts.bmp
BatchRunElWaitingBitmap = pbbwait.bmp
BatchRunElInWorkBitmap = pbbwork.bmp
BatchRunElOKBitmap = pbbok.bmp
BatchRunElErrorBitmap = pbberr.bmp
```

```
; Width of a typical character
CharToGetColWidthPerCharacter= CharToGetColWidth
Language-dependent, see ...\language\patm_xx.ini
; Selection of the tool status bits which are displayed in the
; result list:
; ResultToolStatusColumnsEnable= 1111100100110000
; Bits 1 to 16, bit 1 is the least-significant bit in the tool
status and
; is positioned to the left in this character string in the header
; of the tool status column:
; If ResultToolStatusColumnsHeaderText and
; ResultToolStatusColumnsListText equals "<Empty>",
; the language-dependent values are in the batch list
; of pa_xx.dll.
ResultToolStatusColumnsHeaderText=<Empty> ;language-dependent
; ResultToolStatusColumnsHeaderText=ToolStatusColHeaderText
language-dependent
; Text in the data of the tool status column:
ResultToolStatusColumnsListText= <Empty> ;language-dependent/locale-
specific
; ResultToolStatusColumnsListText= ToolStatusColListText
language-dependent
; Column width for tool identifier
ResultDisplayedNumberOfToolnameCharacters=18
TimeMSecBetweenBatchOrders=1000
; Definitions of individual filters:
1_FindSoftkeyText = F1SK
; Language-specific; prewarning or disabled
1_FindResultHeadlineText = R1HL
; Language-specific; prewarning limit reached or disabled
1_FindCondition = TC_TP8 && 20
; Prewarning bit set (bit 5 of bit 1 to 16 (2 to the power of (5-
1)=16)+
; Disabled bit set (bit 3 (2 to the power of (3-1)=4)
1_FindResultAddColumnBtss = <empty>
1_FindResultAddColumnText = <empty> ; or R1AddCol ; language-
dependent
1_FindResultAddColumnDisplayedNumberOfCharacters=0
1_FindLimitedToCurMagazine=true
; "True", "False" (default); restricted to current magazine
; if called via magazine list.
1_ResultListType =0
0 = Standardliste (default setting), 1 = loading list
1_ReactivatePositioningMode = 2
; Positioning during Reactivate
; 0: Do not position,
; 1: Ask the operator whether to position,
; 2: Always position (default setting)
2_FindSoftkeyText = F2SK ; "disabled"
```

```
2_FindResultHeadlineText = R2HL ; "Tools disabled"
2_FindCondition = TC_TP8 && 4
; Disabled=bit 3 (2 to the power of (3-1)=4)
2_FindResultAddColumnBtss = <empty>
2_FindResultAddColumnText = <empty> ; or R2AddCol
2_FindResultAddColumnDisplayedNumberOfCharacters=0
2_FindLimitedToCurMagazine=False
; "True", "False" (default); restricted to current magazine
; if called via magazine list.
2_ResultListType =0
; 0 = Standard list (default setting), 1 = loading list
2_ReactivatePositioningMode = 0
; 0: Do not position,
; 1: Ask the operator whether to position,
; 2: Always position (default setting)
3_FindSoftkeyText = F3SK ; "Load all"
3_FindResultHeadlineText = R3HL
; Unloading list for all loaded tools
3_FindCondition = A_TOOLMN > 0
; Magazine number of tool greater than 0
3_FindResultAddColumnBtss = <empty>
3_FindResultAddColumnText = <empty> ; or R3AddCol
3_FindResultAddColumnDisplayedNumberOfCharacters=0
3_FindLimitedToCurMagazine=False
; "True", "False" (default); restricted to current magazine
; if called via magazine list.
3_ResultListType =0
; 0 = Standard list (default setting), 1 = loading list
3_ReactivatePositioningMode = 1
; Positioning during Reactivate
; 0: Do not position, 1: Ask the operator whether to
; position, 2: Always position (default setting)
4_FindSoftkeyText = F4SK ; "Unload all"
4_FindResultHeadlineText = R4HL
; Loading list for all unloaded tools
4_FindCondition = A_TOOLMN == 0
; Magazine number of tool equal to 0
4_FindResultAddColumnBtss = <empty>
4_FindResultAddColumnText = <empty> ; or R4AddCol
4_FindResultAddColumnDisplayedNumberOfCharacters=0
4_FindLimitedToCurMagazine=False
; "True", "False" (default); restricted to current magazine
; if called via magazine list.
4_ResultListType =0
; 0 = Standard list (default setting), 1 = loading list
4_ReactivatePositioningMode = 1
; Positioning during Reactivate
; 0: Do not position,
; 1: Ask the operator whether to position,
; 2: Always position (default setting)
5_FindSoftkeyText = F5SK ; "Load identifier"
5_FindResultHeadlineText = R5HL
; "Load list for all tools with load identifier"
5_FindCondition = TC_TP8 && 2048
; (LoadIdentifier=bit12 (2 to the power of (12-1)=2048)
5_FindResultAddColumnBtss = <empty>
5_FindResultAddColumnText = <empty> ; or R5AddCol
5_FindResultAddColumnDisplayedNumberOfCharacters=0
5_FindLimitedToCurMagazine=False
; "True", "False" (default) restricted to current magazine
; if called via magazine list.
5_ResultListType =1
; 0 = Standard list (default setting), 1 = loading list
6_FindSoftkeyText = F6SK ; "Unload identifier"
6_FindResultHeadlineText = R6HL
; "Unload list for all tools with unload identifier"
6_FindCondition = TC_TP8 && 1024
; (UnloadIdentifier=bit11 (2 to the power of (11-1)=1024)
6_FindResultAddColumnBtss = <empty>
6_FindResultAddColumnText = <empty> ; or R6AddCol
6_FindResultAddColumnDisplayedNumberOfCharacters=0
6_FindLimitedToCurMagazine=False
; "True", "False" (default); restricted to current magazine
; if called via magazine list.
6_ResultListType =0
; 0 = Standard list (default setting), 1 = loading list
```

[ShortcutSoftkeysForMagSelect]

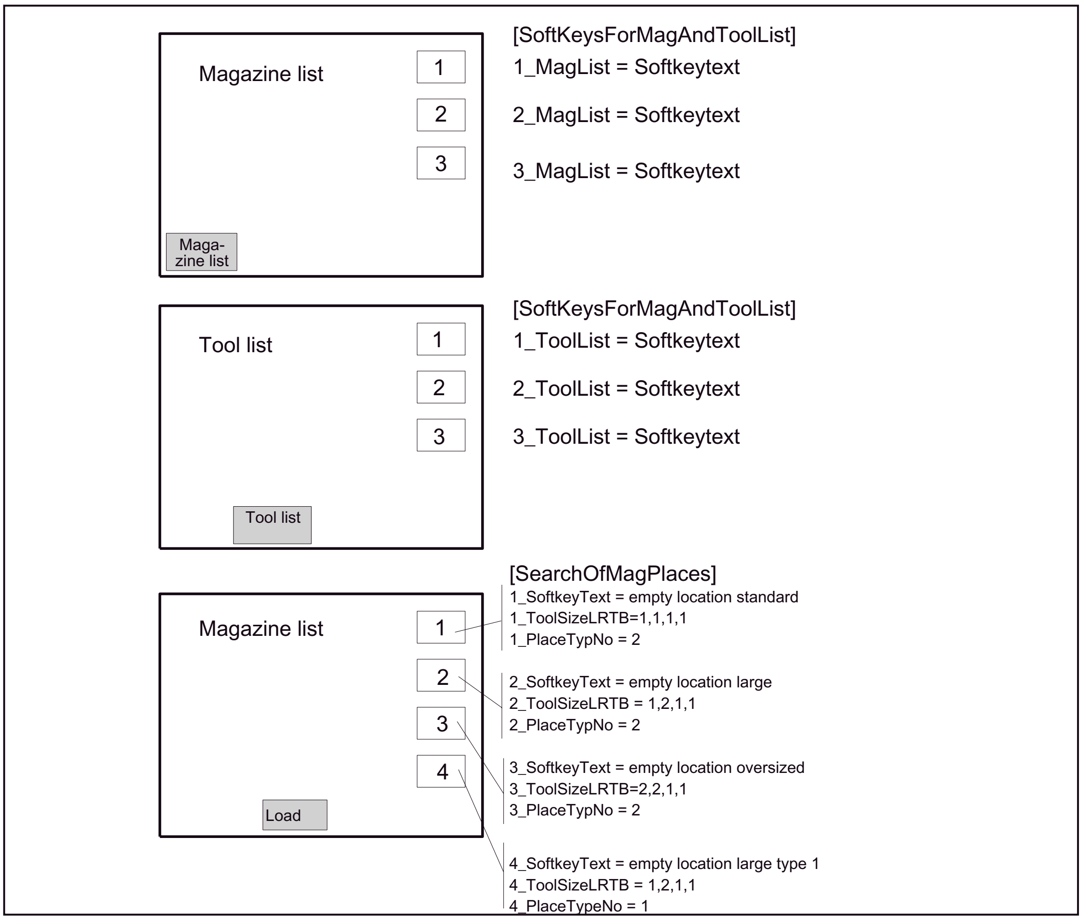
```
; Definition of shortcut softkeys for up to 5 favorite magazines
; per TOA, evaluated if section "[GeneralSettingsForMagAndToolList]"
; entry "MagListMagSelectSoftkey=SelectMag" is set.
; This helps to prevent users from excessive use of "magazine +"
; and "magazine -" softkeys, if a lot of magazines are available.
; You can define up to 5 shortcut softkeys for favorite magazines.
; It is possible to use a shortcut softkey for different magazines,
; if the magazines are in different TOAs.
; Syntax: "magIdent = ShortKeyNumber, AutoReturn"
; Examples: turret15=3,NoAuto<< chain50 =1,Auto<<
; Meaning:
; "magIdent": magazine ident like in $TC_MAP2 or in
; magazine configuration in application maintenance tool
; management.
; "ShortKeyNumber": Number of shortcut, value 1 to 5
; "AutoReturn": stay in magazine selection state or return
; automatically to magazine list state Values "NoAuto<<"
; and "Auto<<".
; magazine "turret15" ($TC_MAP2) can be displayed by shortcut 3,
; you must use "<<" softkey explicitly to leave magazine select
; state.
; magazine "chain50" can be displayed by shortcut 1 and there is
; an automatic return to magazine list state after pressing this
; shortcut softkey. To specify softkey text, use section
; "[ShortcutSoftKeysForMagSelect]" in language dependent
; ini files patm_??.ini.
; chain10 = 1, Auto<<
; turret20 = 2, Auto<<
; turret10 = 3, Auto<<
; chain20 = 5, NoAuto<<
; turret15 = 4, NoAuto<<
```

6.2.3 Configuring the display of softkeys

Softkey texts for the lists

The displays stored behind softkeys 1 to 3 in the magazine and tool lists are defined in the file paramtm.ini.

As the default setting when tool management is selected, the displays appear that have been configured for [1_MagList] and [1_ToolList].



```
[SoftKeysForMagAndToolList]
1_MagList = Softkeytext
2_MagList = Softkeytext
3_MagList = Softkeytext

[SoftKeysForMagAndToolList]
1_ToolList = Softkeytext
2_ToolList = Softkeytext
3_ToolList = Softkeytext

[SearchOfMagPlaces]
1_SoftkeyText = empty location standard
1_ToolSizeLRTB=1,1,1,1
1_PlaceTypNo = 2

2_SoftkeyText = empty location large
2_ToolSizeLRTB = 1,2,1,1
2_PlaceTypNo = 2

3_SoftkeyText = empty location oversized
3_ToolSizeLRTB=2,2,1,1
3_PlaceTypNo = 2

4_SoftkeyText = empty location large type 1
4_ToolSizeLRTB = 1,2,1,1
4_PlaceTypeNo = 1
```

Figure 6-14 Texts for the softkeys

Displaying the displays

Hidden fields can be made visible by scrolling with the cursor keys.

The serial number defined by the input sequence during start-up is displayed in the location type box rather than the name of the location. The screen that is displayed under the 1st vertical softkey in the magazine list is specified after vocabulary word [1_MagList] in file paramtm.ini.

User data

The parameter name and the units can be defined for the displays of the tool and cutting-edge data. How many parameters are displayed depends on the machine data and the number of defined parameters.

[ToolParams]: Tool user data

[ToolEdgeParams]: Tool cutting edge user data

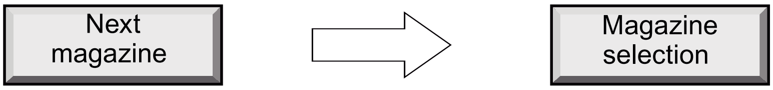
Special characters

Special characters such as ü, ä, ö, ß are entered in ANSI code in order for them to be displayed.

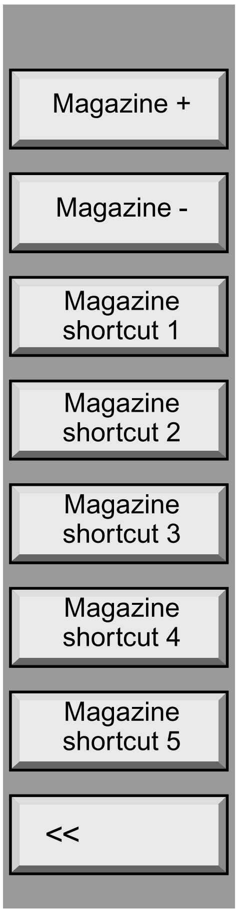
Optional selection of magazines

Up to now, the softkey "Next magazine" could only control the display of the individual magazine lists. If many magazines are present and there are important processes taking place in the magazines with high magazine numbers, this places a burden on the operator.

Via an entry in the ini file the "Next magazine" softkey can be replaced with the "Magazine selection" softkey.



Eight vertical softkeys for faster magazine navigation:



The magazine list is displayed in this status.

(Softkeys 3 to 7 can only be used if the corresponding parameter has been set in the ini file.)

Using the vertical softkeys "Magazine +" and "Magazine -" you can switch to the magazine with the next highest or next lowest magazine number within the TOA of the current channel of the operator panel. (When you are positioned on the last magazine and press "+" you jump to the first magazine; from the first to the last again with "-".)

Using the five vertical softkeys 3 to 7, you can quickly jump to a specific magazine within the TOA of the current channel of the operator panel. The assignment to "Magazine-Ident" and the softkey text must be parameterized in the INI file. When you select a magazine via the vertical softkeys, the magazine list switches immediately to the new magazine.

Press the vertical softkey 8 "<<" to return to the standard magazine list state with the corresponding softkey assignment.

With the five softkeys for rapid magazine selection, you can set an option in the INI file to return to the standard magazine list state automatically. In this case, it is advisable to attach the character string "<<" to the magazine name in the softkey text.

The rapid selection keys support multiple assignment for use in different TOAs and for systems with N:M assignment between HMI Advanced and NCUs. The responsibility of assigning only magazines in different TOAs or different NCUs to the same softkey lies with the person setting the parameters.

Entries in the paramtm.ini file

```
[GeneralSettingsForMagAndToolList]
;In magazine list forms change softkey "Next Mag"
; to softkey "Magazine Selection", to activate vertical softkeys
; in an additional state for magazine selection.
; This helps to prevent users from excessive use of
; "next mag" softkey if a lot of magazines are available.
; You can define shortcut softkeys for up to 5 favorite
; magazines per TOA using section [ShortcutSoftKeysForMagSelect]
; in paramtm.ini and patm_??.ini.

MagListMagSelectSoftkey=NextMag ;default
;MagListMagSelectSoftkey=SelectMag
[ShortcutSoftKeysForMagSelect]
; Definition of shortcut softkeys for up to 5 favorite magazines
; per TOA, evaluated if section "[GeneralSettingsForMagAndToolList]"
; entry "MagListMagSelectSoftkey=SelectMag" is set.
; This helps to prevent users from excessive use of "magazine +" and
; "magazine -" softkeys, if a lot of magazines are available.
; You can define up to 5 shortcut softkeys for favorite magazines.
; It is possible to use a shortcut softkey for different magazines,
; if the magazines are in different TOAs.
; Syntax: "magIdent = ShortKeyNumber, AutoReturn"
; Examples: turret15=3,NoAuto<<
; chain50 =1,Auto<<
; Explanation: "magIdent": magazine ident like in $TC_MAP2 or
; in magazine configuration in application maintenance
; tool management. "ShortKeyNumber": Number of shortcut,
; value 1 to 5 "AutoReturn":
; stay in magazine selection state or return automatically
; to magazine list state Values "NoAuto<<" and "Auto<<".
; magazine "turret15" ($TC_MAP2) can be displayed
; by shortcut 3, you must use "<<" softkey explicitly to leave
; magazine select state.
; magazine "chain50" can be displayed by shortcut 1 and
; there is an automatic return to magazine list state
; after pressing this shortcut softkey.
; To specify softkey text, use section
"[ShortcutSoftKeysForMagSelect]" in language dependent ini files
; patm_??.ini.
chain10 = 1, Auto<<
; turret20 = 2, Auto<<
> ;turret10 = 3, Auto<<
> ;chain20 = 5, NoAuto<<
> ;turret15 = 4, NoAuto<<
```

```
>
language\patm_*.ini:
[ShortcutSoftKeysForMagSelect]
; Softkey text of magazine selection shortcut softkeys.
; Syntax: magIdent=ShortcutSoftkeyText
; Explanation: "magIdent": magazine ident like in $TC_MAP2 or
; in magazine configuration in application
; maintenance tool management.
; "ShortcutSoftkeyText": Softkey text, use double blank to
; indicate wordwrap.
; turret10 = "1-Turret10 <<" // Softkey text
; turret20 = "2-Turret20 <<" // Softkey text
; chain10 = "3-Chain10 <<" // Softkey text
; turret15 = "4-Turret15" // Softkey text
; chain20 = "5-Chain20" // Softkey text
```

Display location status of adjacent locations

In the list displays (magazine list, tool list, working offset list) of the HMI Advanced tool management bits 8 to 16 of the magazine location status can now also be displayed, including the 4 bits "left, right, top, bottom half location reserved".

The NC uses this data when "Adjacent location management" is activated.

The data largely correspond to the NC variable $TC_MPP4.

Until now, bits 1 to 8 of the magazine location status could be displayed in the lists; this expansion now makes it possible to display bits 8 to 16 as well.

The magazine location status bits 8 to 16 can be displayed using the HMI Advanced software; they cannot, however, be changed.

The HMI Advanced software contains a sample parameter assignment (paramtm.ini, paramtm.txt) where display of the additional magazine location status bits is prepared but not activated.

Parameterization

The parameter assignment of the status bit display as column in the individual list views for list displays 1 to 3 (magazine list 1 to 3, tool list 1 bis 3, working offset list 1 to 3) is accordingly expanded to include the nine status bits.

As was the case previously, the individual language-dependent letters for display in the HMI Advanced tool management list displays can also be parameterized in the INI file as an exception, e.g. if the machine operator wants to use different letters or there is no modified language DLL for the locale.

Restrictions

The NC only uses the data "left, right, top, bottom half location occupied/reserved" if "Adjacent location management" is activated. If "Adjacent location management" is not activated in the NC, display can still be activated for the associated magazine location status bits in the HMI Advanced tool management list displays but the displayed values will always be "Bit not set".

Settings in the INI files

paramtm.ini / paramtm.txt:

```
[GeneralSettingsForMagAndToolList]
MagPlaceState_Lang_12345678=<Empty> ; use language-DLL
MagPlaceState_Lang_12345678=12345678_MagPlaceState_Lang
; use patm_*.ini
; The new magazine location status bits can be displayed in each
; list display. The parameters are set in the sections:
"[1_MagList]", "[2_MagList]", "[3_MagList]"
"[1_ToolList]", "[2_ToolList]", "[3_ToolList]"
"[1_ActList]", "[2_ActList]", "[3_ActList]"
; Entries are for example:
12=TC_MPP4_9, 1, TC_MPP4_9 ;PlaceStatus Left,
;Reserved in left half location
13=TC_MPP4_10, 1, TC_MPP4_10 ;PlaceStatus Right,
;Reserved in right half location
14=TC_MPP4_11, 1, TC_MPP4_11 ;PlaceStatus Top,
;Reserved in upper half location
15=TC_MPP4_12, 1, TC_MPP4_12 ;PlaceStatus Bottom,
;Reserved in lower half location
16=TC_MPP4_13, 1, TC_MPP4_13 ;PlaceStatus Bit 13 of 1 to 16
17=TC_MPP4_14, 1, TC_MPP4_14 ;PlaceStatus Bit 14 of 1 to 16
18=TC_MPP4_15, 1, TC_MPP4_15 ;PlaceStatus Bit 15 of 1 to 16
19=TC_MPP4_16, 1, TC_MPP4_16 ;PlaceStatus Bit 16 of 1 to 16
```

patm_gr.ini:

```
[GeneralSettingsForMagAndToolList]
12345678_MagPlaceState_Lang="123456789ABCDEFG" ; // 16 exact
[ListColumnHeaderText]
TC_MPP4_9 = "P" ;PlaceStatus Left,
;Reserved in the left half loc. // 1
TC_MPP4_10 = "P" ;PlaceStatus Right,
;Reserved in the right half loc. // 1
TC_MPP4_11 = "P" ;PlaceStatus Top,
;Reserved in the top half loc. // 1
TC_MPP4_12 = "P" ;PlaceStatus Bottom,
;Reserved in the bottom half loc. // 1
TC_MPP4_13 = "P";PlaceStatus undefined,
;(Wear group disabled) // 1
TC_MPP4_14 = "P" ;PlaceStatus Bit14 of 1 to 16 // 1
TC_MPP4_15 = "P" ;PlaceStatus Bit15 of 1 to 16 // 1
TC_MPP4_16 = "P" ;PlaceStatus Bit16 of 1 to 16 // 1
```

Coding of location status and tool status

Location statuses		Tool statuses	
G	Disabled location	G	Disabled tool
F	Free location	F	Released tool
Z	Reserved for tool in buffer	A	Active tool
B	Reserved for tool to be loaded	M	Measured tool
L	Left half location occupied	V	Prewarning limit reached
R	Right half location occupied	W	Tool is being changed
O	Upper half location occupied	P	Fixed location coded tool
U	Lower half location occupied	E	Tool has been in use
l	Left half location reserved	R	Unloading marking
r	Right half location reserved	B	Loading marking
o	Upper half location reserved	S	Master tool
u	Lower half location reserved		

6.2.4 Parameterizing the display of bitmaps for tools

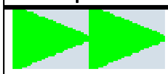
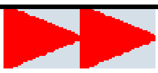
Parameter settings for bitmaps in the list

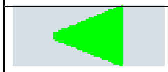
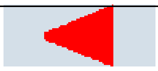
The display of the active tool, the programmed tool and the current location in the magazine list can be freely parameterized, i.e. bitmaps can be inserted in the parameterizable columns of individual lists. The bitmaps can be customized or created from scratch. The display is activated in paramtm.ini.

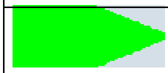
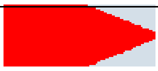
The bitmaps described below are located in the "hmi_adv" directory.

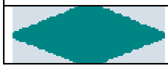
Default bitmap settings

The bitmaps for the current tools are shown in red and those for the programmed tools are shown in green.

Bitmap		Features
		TNo. <> 0; DNo./cutting edge no. <> 0 DLNo. <> 0

		TNo. <> 0;
		DNo./cutting edge no. = 0
		DLNo. = 0

		TNo. <> 0;
		DNo./cutting edge no. <> 0;
		DLNo. = 0

		Current location

User-defined bitmaps can be stored in the "user" directory. They are displayed instead of the standard bitmaps in the lists.

Handling of lists

The columns of the lists in which the bitmaps are to be entered can be set for each list view. The width of the bitmaps is set in characters for the entire highlighted areas. The width of the column is automatically increased by the value set.

Bitmaps overwrite mutually when displayed in the same column and line. The highlighting at the top represents the current tool, the highlighting below represents the programmed tool and the highlighting at the bottom represents the current location. Hidden bitmaps are not displayed.

Note

In multi-line magazine and tool lists, the highlighting is entered in the cutting edge line when the current/programmed DNo./cutting edge no. <> 0. The same applies in working offset lists for DLNo. <> 0 for the DL rows. Since only cutting edges can be displayed in the views of the working offset lists, the highlighting only appears if the current/programmed DNo./cutting edge no. <> 0.

The current magazine location is only highlighted in the magazine list views. Highlighting only appears in the normal magazine display and not in the buffer magazine display.

Setting the bitmap parameters

By default, the bitmaps are not entered in paramtm.ini and are not displayed. If the bitmaps are to be displayed in the lists, you will have to make some changes to the parameter file. One entry is required for each bitmap.

Entries in paramtm.ini:

[GeneralSettingForMagAndToolList]

```
; Width of the bitmap display
; Unit: number of characters
WidthOfActBitmapsInCharacters=5
; Name of the bitmap for the current tool /DNo./DL,
; where D<>0 and DL<>0
ActToolBitmap=paat.bmp
; Name of the bitmap for the current tool/DNo.,
; if the current cutting edge D=0.
ActToolZeroDBitmap=paatd0.bmp
; Name of the bitmap for the current tool /DNo./DL,
; if the current DL=0.
ActToolZeroDLBitmap=paatdl0.bmp
; Name of the bitmap for the programmed tool /DNo./DL,
; where D<>0 and DL<>0
ProgToolBitmap = papt.bmp
; Name of the bitmap for the programmed tool/DNo.,
; if the current cutting edge D=0.
ProgToolZeroDBitmap = paptd0.bmp
; Name of the bitmap for the programmed tool /DNo./DL,
; if the current DL=0.
ProgToolZeroDLBitmap = paptdl0.bmp
; File name of the bitmap for the current magazine location
ActPlaceBitmap = paap.bmp
```

[1_MagList]

```
; Columns in which highlighting (bitmaps) is to be
; displayed:
ShowActToolCol = 1
ShowProgToolCol=1
ShowActPlaceCol=1
```

6.2.5 Example: Customized settings for the machines

Configuring the tool management displays in the paramtm.ini file

The tool management operator interface can be customized with the file paramtm.ini:

- Modify the structure and layout of the lists.
- Apply specific default values.
- Protect or deactivate functions via access rights.

All the functions and possibilities for tool management are listed in the paramtm.txt file on HMI Advanced.

At installation and start-up, the operator can decide which functions are required for a specific machine. Values and to some extent also functions can be preset to allow for convenient, user-friendly operation.

Examples

Example 1:

- The tool data is to be automatically deleted when unloaded (magazine list only).
- The tool list function is not used.
- The function tool catalog and cabinet is not used.

The parameters can be assigned as follows:

```
...
[TMMODES]
...
DELETE_TOOL_ON_UNLOAD=1 The tool data is automatically deleted
on unloading.
...
[ACESSLEVEL]
```

```
...
SKTLLIST=2 The tool list can only be activated with the manufacturer
password, i.e. it is blocked in normal operation.
SLTOOLCAB=2 The tool catalog and cabinet SKTOOLCAT=2 can only be activated
with the manufacturer password, i.e. they are blocked in normal operation.
...
```

Example 2:

- The tool data is not deleted at unloading, but remain in the tool list (in NCK). The data can be used for loading tools.
- The function tool catalog and tool cabinet is not used.

The parameters can be assigned as follows:

```
...
[TMMODES]
...
DELETE_TOOL_ON_UNLOAD=0 The tool data is not deleted
on unloading.
...
[ACESSLEVEL]
...
SKTLLIST=7 The tool list can always be called.
SLTOOLCAB=2 The tool catalog and cabinet SKTOOLCAT=2 can only be activated
with the manufacturer password and are blocked for the user.
...
```

Example 3:

- The tool data is to be automatically deleted at unload in the magazine list.
- The function tool catalog and tool cabinet is used.

The parameters can be assigned as follows:

```
...
[TMMODES]
...
DELETE_TOOL_ON_UNLOAD=1 The tool data is deleted
on unloading.
...
[ACESSLEVEL]
...
SKTLLIST=2 The tool list is only active with the manufacturer password.
SLTOOLCAB=7 The tool catalog and tool cabinet
SKTOOLCAT=7 can be called (are not locked)
...
```

Note

Access authorization

If access rights have been assigned for functions and the protection level is "less" than that assigned, then the softkey is not displayed in the operator interface and the function cannot be called.

This applies to all functions. If for example, the "Tool cabinet" function is barred from operation, then its softkeys are not displayed.

6.3 Language-dependence for user-defined name

6.3.1 Language-dependent name for magazines

Function

The magazine names are configured by the user in the Magazines screen via the tool management start-up tool (IW). This is why the assigned names are contained in the tool management database and after loading a magazine configuration also in the NCK but not in a language DLL. In previous versions, this meant that they were not available in different languages.

The new functionality allows the user to create the magazine names in different languages/locales.

You can achieve this by entering name texts in the language-specific tool management INI files for the magazines configured in the tool database.

There are two names for each magazine:

- the standard name (also known by the NCK) which is used for functional operation
- an associated language-specific name which is displayed on the operator interface.

If the user does not assign a language-specific name, the standard name from the NCK is displayed in the tool management; and the standard name from the tool database is displayed in the startup tool.

Entries in the language-specific INI files

Users must make the entries described here themselves. They are not written to the to the INI files by HMI Advanced.

The language-specific INI files reside under ../hmi_adv/language.

File	patm_xx.ini
Section	[Magazine_VISName]
Entry	Standard name="Language-specific text"
Example:	"[Magazine_VISName]"
	Chain1="Chain magazine 1"
	Turret1="Turret 1"
	...

For newly entered texts from the INI files to be activated in the display, you need to change the language setting or start HMI Advanced again.

Display magazine names in the HMI tool management screens

The language-specific names of the magazines are displayed in all the tool management screens and the tool management start-up screens. If there are no entries are the corresponding INI files, the standard names from the NCK (in the tool management) or from the tool database (in the startup tool) are displayed.

Affected screens/functions:

- Tool management:
  – Magazine list
  – Magazine selection
- Tool management startup:
  – Magazines
  – Buffer
  – Loading locations
  – Magazine configuration

6.3.2 Language-specific names for buffers

Function

The buffer locations and their names are entered by the user via the tool management startup tool (IW) in the Buffers screen. This is why the names are specified in the tool management database and not in a language DLL. In previous versions, this meant that they were not available in different languages.

Until now two options were available for displaying the buffer locations in the tool management (not tool management start-up):

- Display names from the tool database (paramtm.ini, [TMMODES] NameOfBufferPlaceFrom=DB)
- Display the type of buffer location from the language DLL plus the associated index. In other words, for example, Spindle1, Spindle2 or Gripper2, etc. (NameOfBufferPlaceFrom=DLL)

The new functionality allows the user to create the buffer location names in different languages/locales.

For this display it is necessary to select **Display name from the database (NameOfBufferPlace From=DB)**.

The user must enter a corresponding name text in the language-specific tool management INI files for each buffer location configured in the database.

There are two names for each buffer location: The standard name which is used for internal processing (tool database) and an associated language-specific name which is displayed on the operator interface.

If the user does not assign a language-specific name, the standard name from the database is displayed (as was the case up to now).

The tool management startup tool does not evaluate the entry NameOfBufferPlaceFrom and always uses the name from the tool database and the language-specific INI files.

Entries in the language-specific INI files

Users must make the entries described here themselves. They are not written to the to the INI files by HMI Advanced.

The language-specific INI files are called patm_xx.ini and can be found under ../hmi_adv/language. The user-defined patm_xx.ini files are found under .../user/language.

The INI file paramtm.ini and the associated description file paramtm.txt reside under ../hmi_adv. The user-defined file paramtm.ini resides under .../user. As it is the default setting, there is no need to make an entry in the user-defined paramtm.ini.

File	paramtm.ini
Section	[TMMODES]
Entry	NameOfBufferPlaceFrom=DB (default)
and	
File	patm_xx.ini
Section	[BufferPlace_VISName]
Entry	Standard name="Language-specific text"
Example:	[BufferPlace_VISName]
	Spindle1="Main spindle"
	Gripper1="1st gripper"
	...

For newly entered texts from the INI files to be activated in the display, you need to change the language setting or start HMI Advanced again.

Display buffer names in the HMI tool management screens

The language-specific names of the buffer locations are displayed in all the relevant tool management screens and tool management start-up screens. If there are no entries in the corresponding INI files, the standard name from the tool database is displayed.

Affected screens/functions:

- Tool management: Magazine list, with display of buffer
- Tool management startup: Buffer

6.3.3 Language-dependent name for loading locations

Function

The loading locations and their names are entered by the user via the tool management startup tool (IW) in the Loading locations screen. This is why the names are specified in the tool management database and not in a language DLL.

You can achieve this by entering name texts in the language-specific tool management INI files for the loading locations configured in the database.

There are two names for each loading location:

- the standard name which is used internally (tool database)
- an associated language-specific name which is displayed on the operator interface.

If the user does not assign a language-specific name, the standard name from the tool database is displayed.

This also applies to the first location in the loading magazine which is automatically assigned:

The tool management start-up tool (IW) automatically creates an entry in the tool database for the first location in the loading magazine. This occurs at first access to the loading locations screen with an original database. This location must always exist, therefore it is not possible to delete it.

It is assigned the internal standard name "FirstLoadingPoint" with the following characteristics:

- The default setting for all patm_xx.ini files contained in the scope of supply includes an entry in [LoadLocation_VISName] for the location type "FirstLoadingPoint".
- In the Loading locations screen for the tool management startup tool (IW) the language-specific text from patm_xx.ini is also displayed for the first loading location in the "Name" selection box.

Already existing databases in systems that have already been operating for some time:

In older HMI versions, this 1st loading point was called "Loading point for spindle" or "Loading point manual" (in the language set up at this time).

If it is detected on first startup of the tool management or associated startup tool (IW) with the functionality described here, then the existing name in the database can be replaced with "FirstLoadingPoint".

Entries in the language-specific INI files

Users must make the entries described here themselves. They are not written to the to the INI files by HMI Advanced.

Exception: FirstLoadingPoint="Loading point manual"

The language-specific INI files are called patm_xx.ini and can be found under ../hmi_adv/language. The user-defined patm_xx.ini files are found under .../user/language.

File	patm_xx.ini
Section	[LoadLocation_VISName]
Entry	Standard name="Language-specific text"
Example:	[LoadLocation _VISName] FirstLoadingPoint="Loading point manual" Loading station1="Main loading station" ...

For newly entered texts from the INI files to be activated in the display, you need to change the language setting or start HMI Advanced again.

Display loading locations in the HMI tool management screens

The language-specific names of the loading locations are displayed in all the relevant tool management screens and tool management start-up screens. If there are no entries in the corresponding INI files, the standard name from the tool database is displayed.

Affected screens/functions:

- Tool management:
  - Load
  - Unload
  - Empty location search
  - Positioning
- Tool management startup: Loading locations

6.3.4 Language-specific names for location types

Function

The location types and their names are entered by the user via the tool management startup tool (IW) in the "Location types" screen. This is why the names are specified in the tool management database and not in a language DLL. In previous versions, this meant that they were not available in different languages.

The new functionality allows the user to create the location type names in different languages/locales.

You can achieve this by entering name texts in the language-specific tool management INI files for the location types configured in the database.

There are two names for each location type:

- The standard name which is used internally (tool database) and
- an associated language-specific name which is displayed on the operator interface.

If the user does not assign a language-specific name, the standard name from the database is displayed.

The location type "standard" is contained in the original database shipped with the tool management. The following special handling applies for this location type:

- The default setting for all patm_xx.ini files contained in the scope of supply includes an entry in [Placetype_VISName] for the "standard" location type.
- In the Location types screen for the tool management startup tool (IW) the language-specific text from patm_xx.ini is also displayed for the "standard" location type in the "Name" selection box.

Entries in the language-specific INI files

Users must make the entries described here themselves. They are not written to the to the INI files by HMI Advanced.

Exception: Standard="Location type standard".

The language-specific INI files are called patm_xx.ini and can be found under . ./hmi_adv/language. The user-defined patm_xx.ini files are found under ../user/language.

File	patm_xx.ini
Section	[Placetype_VISName]
Entry	Standard name="Language-specific text"
Example	[Placetype_VISName]
	Standard="Location type standard".
	SmallPlaceType="small"
	...

For newly entered texts from the INI files to be activated in the display, you need to change the language setting or start HMI Advanced again.

Display location type names in the HMI tool management screens

The language-specific names of the location types are displayed in all the tool management screens and the tool management start-up screens. If there are no entries in the corresponding INI files, the standard name from the tool database is displayed.

Affected screens/functions:

- Tool management:
  - Tool details
  - Tool new
  - Tool catalog
  - Tool cabinet
  - Empty location search
- Tool management startup:
  - Magazine configuration
  - Location types

Tool catalog/tool cabinet

The standard name of the assigned location type is maintained and internally processed for each tool in the tool catalog/cabinet.

The language-specific name is displayed in the tool catalog/cabinet screens for the location type. If there is no language-specific name, the standard name from the tool database is used instead.

Code carrier

The name of the magazine location type which is written via the dialog variable T8 to the code carrier chip always corresponds to the standard name of the magazine location type from the tool database.

Alternatively, the name of the magazine location type can be written to the code carrier via dialog variable T12. T8 and T12 are also accepted. When the chip is read in, a cross check is performed to associate magazine location type name and number.

6.4 Job processing of tools

Prerequisite

The settings for batch processing of tools are contained in paramtm.ini and language\patm_xx.ini in the section [BatchTools].

The filter only works if bit 4 is set in MD 18080: TOOL_MANAGEMENT_MASK.

Up to 6 search filters can be defined. The following can be specified for each filter: Softkey text, results title, search criteria, selection of the type of results list and additional data.

The file ...user\paramini.out contains error messages which occurred when reading in the parameter assignment.

6.4.1 Settings that apply for all filters at the same time

Colors for the results list

The colors for the results list are customizable. When setting colors, please avoid color combinations that are difficult to read or too bright.

The colors have to be specified for the following list elements:

- Non-selected text
- Non-selected background
- Cursor-selected text
- Cursor-selected background
- Job-selection and cursor-selected text
- Job-selection and cursor-selected background
- Job-selection and cursor-selected text
- Job-selection and cursor-selected background

The parameter settings consist of eight colors separated by comma. A hexadecimal value consisting of 8 characters is assigned to each color.

The hex value has the following syntax:

SSBBGGRR where SS=System, BB=Blue, GG=Green, RR=Red

Examples for colors:

- 80 00 00 08: Windows text
- 80 00 00 05: Windows background
- 80 00 00 0E: Highlighted Windows text
- 80 00 00 0D: Highlighted Windows background
- 80 00 00 09: Windows active window, header text
- 80 00 00 02: Windows active window, header background
- 00 FF FF 00: Light blue

- 00 00 FF 00: Green
- 00 FF 80 00: Blue green

Examples for setting colors, see [BatchTools], "General settings which apply to all filters" in paramtm.txt.

Bitmaps for the status display of the individual job elements

The user can replace the names of the bitmaps or the bitmaps themselves with custom bitmaps. The custom bitmaps are stored in the "user" directory.

Examples for setting bitmaps, see [BatchTools], "General settings which apply to all filters" in paramtm.txt.

Example:

```
BatchFilterElBUnTUnBitmap = pbfbutu.bmp

BatchFilterElBUnTSeBitmap = pbfbuts.bmp

BatchFilterElBSeTUnBitmap = pbfbstu.bmp

BatchFilterElBSeTSeBitmap = pbfbsts.bmp

BatchRunElWaitingBitmap = pbbwait.bmp

BatchRunElInWorkBitmap = pbbwork.bmp

BatchRunElOKBitmap = pbbok.bmp

BatchRunElErrorBitmap = pbberr.bmp
```

Width of a typical character

For each language you can specify a character whose width is used as the basis to calculate column widths from a specified number of characters. A wide character should be entered here, in Europe typically an "X" or "A".

Example:

Paramtm.ini, [BatchTools]:

```
CharToGetColWidthPerCharacter=CharToGetColWidth ;language-specific
```

language\patm_gr.ini, [BatchTools]:

```
CharToGetColWidth="A"
```

Column width for tool identifier

You can set the column width for the tool identifier as as a rule the full number of 32 characters is not used.

Example:

Paramtm.ini, [BatchTools]:

```
ResultDisplayedNumberOfToolnameCharacters=18
```

Tool status bits

You can set which tool status bits are to be displayed in the results list. Language-specific letters can also be specified in bits for header and list lines.

Example:

Paramtm.ini, [BatchTools]:

ResultToolStatusColumnsEnable= 1111100100110000

1: display, 0: do not display. Bit 1 to 16,

bit 1 is the least-significant bit in the tool status and is positioned to the left in this character sequence

ResultToolStatusColumnsHeaderText=<Empty>

Text in the header for the tool status column, language-specific

ResultToolStatusColumnsListText=<Empty>

Text in the data for the tool status column, language-specific

ToolStatusColHeaderText="123456789ABCDEFG"

Header; Bits 1 to 16, bit 1 is the least-significant bit in the tool status and is positioned to the left in this character string.

ToolStatusColListText="123456789ABCDEFG"

Data; Bits 1 to 16, bit 1 is the least-significant bit in the tool status and is positioned to the left in this character string.

User authorizations

You can set the user rights for the softkeys involved in the paramtm.ini file, [ACCESSLEVEL] section, entries "SKB...").

Example:

[ACCESSLEVEL]

SKBATCH=7; Softkey filter lists

SKFILTER1=7; Softkey Filter1

SKFILTER2=7; Softkey Filter2

SKFILTER3=7; Softkey Filter3

SKFILTER4=7; Softkey Filter4

SKFILTER5=7; Softkey Filter5

SKFILTER6=7; Softkey Filter6

SKBMAGFILTER=7; Softkey magazine selection

SKBATREACT=7; Softkey batch function "Reactivate"

SKBATTOCABIN=7; Softkey batch function "In cabinet"

SKBATDELTOOL=7; Softkey batch function "Delete"

SKBATUNLOAD=7; Softkey batch function "Unload"

SKBFILTERACT=7; Softkey batch function "Update"

SKBATLOAD=7; Softkey batch function "Load"

SKBATLIST=7; Softkeys for controlling the job processing

6.4.2 Settings that are valid for one filter at a time

Search criteria

The search criteria are set in "n_FindCondition" (with n=1 to 6) for each filter.

A maximum of 8 part conditions are permissible, separated by ",". They are ANDed. No data may occur more than once in the part conditions.

Each part condition consists of three parts:

1. Datum for which the condition applies
2. Conditions
3. Comparison Value

Example:

```
1_FindCondition = TC_TP8 && 20, A_TOOLMN > 0
```

Filter 1 filters tools in the NC which fulfill the following condition:

```
( (prewarning bit set (bit5 of bit1 to 16 (2 to the power of (5-
1)=16))))
 OR
 (disabled bit set (bit3 (2 to the power of (3-1)=4)))
)
AND
( (magazine number > 0 means "loaded tool"))
```

Filter criterion

The following data in the NC can be a filter criterion:

Tool data

TC_TP1: Duplo number

TC_TP2: Tool identifier

TC_TP3: Tool size in half locations left

TC_TP4: Tool size in half locations right

TC_TP5: Tool size in half locations top

TC_TP6: Tool size in half locations bottom

TC_TP7: Tool location type

TC_TP8: Tool status

TC_TP9: Monitoring type

TC_TP10: Replacement tool search

TC_TP11: Tool information/replacement tool sequence

A_TOOLMN: Magazine number

A_TOOLMLN: Magazine location number

P_TOOLND: Number of cutting edges

Tool cutting edge parameters

TC_DP1 to TC_DP24

Tool cutting edge monitoring parameters

TC_MOP1: Tool life warning limit

TC_MOP2: Tool life actual value

TC_MOP3: Pre-warning limit for count

TC_MOP4: Actual value for workpiece count

TC_MOP5: Prewarning limit for wear

TC_MOP6: Remaining wear

TC_MOP11: Setpoint for tool life

TC_MOP13: Setpoint for workpiece count

TC_MOP15: Setpoint for wear

Tool OEM data

TC_TPC1 to TC_TPC10

Tool OEM data must be activated on the NC and the numbers must be permissible on the NC.

Tool cutting edge OEM data

TC_DTPC1 to TC_DTPC10

The tool cutting edge OEM data must be activated on the NC and the numbers must be permissible on the NC.

Tool cutting edge monitoring OEM data:

TC_MOPC1 to TC_MOPC10

The tool cutting edge monitoring OEM data must be activated on the NC and the numbers must be permissible on the NC.

Conditions

The following conditions can be used:

==	Equal to
<	less than
>	Larger
<=	Less than or equal to
>=	Greater than or equal to
&&	bit-wise AND between screen and data; only permissible for operands of type WORD and DOUBLEWORD; the individual result bits are ORed.

If this condition is applied to the tool status, you can locate tools with specific set (AND) tool status bits. If several set bits are queried at the same time, just one set bit is sufficient for the tool to appear in the hit list.

Comparison Value

For string operands (e.g. tool identifier) "==" is the only permissible comparison operator. The following value ranges apply:

String: with TC_TP2, maximum of 32 characters, no blanks before or after

0 ... 65535: for the other TC_TP data

Double: for all other data

Additional data

Max. one column with additional data can appear for each filter in the results list.

There are three settings for each filter:

- n_FindResultAddColumnText

  Header text for column or reference to the header text with language-specific settings/locales.
- n_FindResultAddColumnDisplayedNumberOfCharacters

  Column width in characters
- n_FindResultAddColumnBtss

  OPI item acc. to OPI documentation (mmc2\btss_gr.hlp).

Example 1:

Paramtm.ini, [BatchTools]:

1_FindResultAddColumnText=<empty>

; No additional column

1_FindResultAddColumnText=R1AddCol

; Language-specific

1_FindResultAddColumnDisplayedNumberOfCharacters=8

1_FindResultAddColumnBtss="/Tool/User/data[u#TOA#,c3,#TNO#] (l"!l%ld")"

language\patm_gr.ini, [BatchTools]:

R1AddCol="Additional data search 1"

Example 2:

"/Tool/User/data[u#TOA#,c2,#TNO#](l"!d%.#RES#lf")"

Tool OEM parameter2, floating point representation, standard number of decimal places

Example 3:

"/Tool/User/data[u#TOA#,c3,#TNO#](l"!l%ld")"

Tool OEM parameter3, integer representation

Example 4:

"/Tool/MagazineDescription/userData[u#TOA#,c#MAG,1](l)"

Magazine OEM parameter 1

Placeholder

The following placeholders are permissible:

#TOA#	TOA number (of the current channel)
#TNO#	Internal T number (of the tool found)
#MAG#	Magazine number (of the tool found)
#RES#	Standard value for the number of decimal places

Placeholders are substituted by the data for the current tool or by general settings.

Max. 1 OPI item is permitted.

OPI multiple variable accesses are generated internally from the OPI item.

"(|)" is entered in front of the result data to generate the data separation character "|".

The OPI item must be enclosed by " " especially if it contains formatting information (e.g. "!I%ld") enclosed by " ".

The syntax is not checked by the operator interface software. The person setting the parameters is responsible for the correct syntax.

Note

If errors are made in the parameter settings, the result list can no longer be displayed or the secondary faults can occur.

Filter name and softkey text

You can set one name and one softkey text in each language for each filter.

There are two settings for each filter:

- n_FindResultHeadlineText

  Text for filter name or reference
- n_FindSoftkeyText

  Softkey text for the filter or reference (a double blank in the text defines a new line)

Example:

Paramtm.ini, [BatchTools]:

1_FindResultHeaderText = R1HL

1_FindSoftkeyText = F1SK

language\patm_gr.ini, [BatchTools]:

R1HL = "prewarning limit reached or disabled"

F1SK = "prewarning or disabled"

Result list type

You can select the result list type for each filter. determines which job functions are available for each softkey.

There is one setting for each filter: n_ResultListType

Result list type, value range:

0:	Standard list (default) for unload, delete, into the cabinet, reactivate
1:	Load list for loading, reactivate

Example:

Paramtm.ini, [BatchTools]:

1_ResultListType = 0 ;0 = standard list

Filter restricted to one magazine

For each filter you can select whether it is restricted to a specific magazine. This should be visible in the filter name.

There is one setting for each filter: n_FindLimitedToCurMagazine

Filter restricted to a specific magazine, value range:

TRUE:	hit list restricted to current magazine
FALSE:	(default) hit list is not restricted to the current magazine

Example:

Paramtm.ini, [BatchTools]:

1_FindLimitedToCurMagazine=true

;"True", "False"; can be restricted to current magazine

Positioning during Reactivate

For each filter you can select whether the job function can be selected, whether with job function "Reactivate" the tool is to be positioned in a loading point.

There is one setting for each filter: n_ReactivatePositioningMode

Position at reactivate, value range:

0	Do not position
1	For each complete job ask the operator whether to position
2	Always position (default setting)

Example:

Paramtm.ini, [BatchTools]:

1_ReactivatePositioningMode=2 ;always

Parameter for PI TSEARCH

Note

Siemens reserves the right to withdraw support in future versions.

The parameters for the PI TSEARCH used for filtering can be specified for each filter.

This setting is very sensitive to errors. It does not support insertion of blanks; the number of places must be strictly adhered to; the character string must be enclosed by " ".

There is one setting for each filter: n_FindPiSearchPar

- 8 parameters for PI SEARCH
- #Mag# can be used as a placeholder for the magazine setting. Constant, five-digit magazine numbers can also be specified for the from/to magazine range instead of the placeholder.
- If the value in the 8th parameter is set to "2", filter criteria for cutting edge specific data will also be used correctly for multi-point cutting tools (from NCK version NCK.P6_43 and NCK.P5_20.4).

Example:

Paramtm.ini, [BatchTools]:

```
1_FindPiSearchPar="#Mag#,-0001,#Mag#,-0001,00000,00001,1,2"
```

6.5 Grinding tools and tool-specific grinding data

Overview

The HMI Advanced tool management is modified so that the "tool-specific grinding data" of grinding tools is displayed and can be edited.

This data is exchanged with the NC via the OPI block TG; it largely corresponds to the NC variables $TC_TPG1 to $TC_TPG9.

For more information on softkey extension in the tool details main screen, tool details cutting edge data screen and tool detail monitoring data screen, see:

References: Operating Manual HMI-Advanced

Setting parameters for the default values

The parameters for the default values for the tool-specific grinding data when creating tools are set in the "paramtm.ini file, section [DEFAULT_ SETTINGS].

Description

```
[DEFAULT_SETTINGS]
; !!! Default setting of grinding-specific tool data at creation:
; !!! If the machine is operating with inch/mm
; !!! ($MN_CONVERT_SCALING_SYSTEM=1) conversion,
; !!! the length unit must be specified !!!
; The following default values (TOOLGRIND..., if affected
; by the length unit) are specified on the basis
; of this length unit:
; 0 = mm (default)
; 1 = inch
TOOLGRIND_Default_Length_Unit=0
; Spindle number (as $TC_TPG1)
TOOLGRINDspinNoDress=1
; Chain rule (as $TC_TPG2)
TOOLGRINDconnectPar=1050629
;1050629 binary: 0000 0000 0001 0000 0000 1000 0000 0101
;Bit0 =1 = Type
;Bit2 =1 = Geo-L1
;Bit11 =1 = Wear-L1
;Bit20 =1 = Base-L1
;Minimum wheel radius (as $TC_TPG3)
TOOLGRINDminToolRadius=0
;Minimum wheel width (as $TC_TPG4)
TOOLGRINDminToolWide=0
;Current width of grinding wheel (as $TC_TPG5)
TOOLGRINDactToolWide=0
; Maximum grinding wheel speed (as $TC_TPG6)
TOOLGRINDmaxRotSpeed=0
```

```
;Maximum grinding wheel peripheral speed (as $TC_TPG7)
TOOLGRINDmaxTipSpeed=0
;Inclination angle of inclined wheel (as $TC_TPG8)
TOOLGRINDinclAngle=0
;Compensation parameter for grinding wheel peripheral speed (as
$TC_TPG9)
TOOLGRINDparamNrCCV=3
```

Note

The HMI Advanced function "Change tool" used up to now is not modified. Therefore, with grinding tools too, when the tool type is changed, most tool data is set by the HMI to "0".

The grinding-specific tool data is not set to "0"; instead it is processed by the NCK.

6.6 Inch/metric setting

Overview

The tool database (tool cabinet and tool catalog) and code carrier support machining in inch or mm measurements.

The behavior with respect to length units is set for the tool database or the code carrier in the paramtm.ini file in the [TMMODES] section via the DATABASE_LENGTH_UNIT and CODECARRIER_LENGTH_UNIT entries.

Tool database

NCK with inch/mm data conversion:

```
[TMMODES]
; !! NOTICE:
; The settings described here are only relevant
; if the machine data $MN_CONVERT_SCALING_SYSTEM=1 is set
; in the NCK.
; The entry for DATABASE_LENGTH_UNIT is only analyzed if
; there is no unit entered in the tool database yet.
; This means once in a normal case scenario!
; -1 means no setting for inch/metric in the tool
; database (default).
; The setting $MN_CONVERT_SCALING_SYSTEM=1 in the NCK means
; that inch/metric conversion is to be used.
; Therefore the user must determine the unit of the data in
; the tool database. With the setting -1 this
; does not occur, so all softkeys for database activities
; are disabled.
; 0 means mm.
; When a tool database is opened for the first time by a tool
management
; with inch/mm differentiation, the data concerned appear
; in the tool database in mm.
; An entry is added to the database to show that
; the tool data in the database is in
; mm.
; For data transfers between the database and NCK
; this is taken into account accordingly.
; 1 means inch
; When a tool database is opened for the first time by a tool
management
; with inch/mm differentiation, the data concerned appear
; in the tool database in inches.
; An entry is added to the database to show that
; the tool data in the database is in
; inches. For data transfers between the database and NCK
; this is taken into account accordingly.
```

```
DATABASE_LENGTH_UNIT=-1
; If DATABASE_LENGTH_UNIT is assigned the setting 0 or 1, the
; following occurs: The first time a new tool management with
; inch/metric differentiation is started up, a new entry is added to
; the database defining the unit for the database for
; future use.
```

The unit entered in the database then applies for processing. The data is written to and read out from the tool cabinet in the specified unit.

Name	Type	Size
InfoKey_Name	Text	255
InfoKey_Index	Integer	2
Info_String	Text	255
Info_Num	Double	8

Table 6- 1 Contents of table for data in mm or inch:

InfoKey_Name	InfoKey_Index	Info_String	Info_Num
"BasicLenUnit"	0	"mm"	0
"BasicLenUnit"	0	"inch"	1

Tool management active

If tool management is in use and activated, you can switch between inch/mm units of measurement as follows:

1. Via softkey e.g. in the machine.

   Then run the tool management. When selected again the data is displayed in all screens in the new unit.

2. Switchover by changing the machine data 10240. $MN_SCALING_SYSTEM_IS_METRIC and PowerOn reset or machine control panel reset on startup.

   Then run the tool management. When selected again the data is displayed in all screens in the new unit.

3. Switchover by changing the machine data 10240. $MN_SCALING_SYSTEM_IS_METRIC and PowerOn reset.

   During the reset process switchover is according to tool management.

   – Tool detail displays: the focus returns to the associated list display

   – List displays: The data is updated accordingly. If necessary (due to reset) a screen change takes place

   – Catalog/cabinet displays with affected data: The display screen is exited with <Cancel>.

4. Switchover by changing the machine data 10240. $MN_SCALING_SYSTEM_IS_METRIC.

   Then call tool management and Power ON reset in the already selected tool management. This corresponds to point 3.

5. Switchover by changing the machine data 10240. $MN_SCALING_SYSTEM_IS_METRIC.

   Then call tool management and machine control panel reset in the already selected tool management.

   – Tool detail displays and list displays: As the individual values are immediately written after input and are constantly updated in these displays, the data is immediately displayed in the new unit.

   – Catalog/cabinet displays with affected data: As the entered data is only written in full to the database when a softkey with a "Save function" (e.g. <OK>, <New cutting edge>) is activated, the new unit only becomes active after this type of action.

See also

Description of the code carrier data (Page 258)

6.7 Start-up of code carrier

6.7.1 Description of the code carrier data

Overview

To the code carrier system: see the functional descriptions for the individual tool identification systems.

A code carrier system is connected via the RS-232 interface, for example. If the machine has its own code carrier system (tool identification system), then this system must also be started up separately.

Note

From SW 6.3 code carrier systems can only be operated via SinTDC and no longer directly.

WToolIdSys = SinTDC

See also: /FBTC/ SINUMERIK Tool Data Communication SinTDC.

In order for the code carrier system to be activated from the tool management, it must be entered in the file ...\user\mmc.ini.

Settings for inch/metric units of measurement and validation of tool status bits are made in the file ...\user\paramtm.ini.

Note

None of the INI files in the "mmc2" directory may be modified.

An INI file is associated with every manufacturer-specific server ("exe file") Manufacturer-specific settings are made in this INI file via the code carrier system. The parameters it contains are described in the documentation from the code carrier manufacturer or the relevant Description of Functions for the tool identification system.

Activating the code carrier system

The connected code carrier system (e.g. WToolIdSys=Ballu) is activated in the file ...\user\mmc.ini.

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[ToolMgmt]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
WToolIdSys=0 ; or Ballu
; Identifier for code carrier system
; 0 means :"No code carrier active"
; Specify manufacturer name (only first 5 characters!)
; Ballu means: Code carrier from manufacturer Balluff is active
WToolIdSysKonv=wkonvert.txt
; Name of the conversion file used for the
; code-carrier format.
; The file resides in directory ...\add_on or ...\user.
[TIS]
; Tool Identification System
; EOT for code-carrier data
TIS_EOT=0x2F2F
; The end identifier for data on the code carrier has to be
; entered here.
;
```

Setting for inch/metric in the paramtm.ini file

The excerpts printed below are found in the paramtm.ini. In newer software versions, the paramtm.ini comments have been summarized in paramtm.txt.

As a general rule, the desired entries are made in ...\user\paramtm.ini so that they are retained during the next HMI software update.

If the paramtm.ini or paramtm.txt of the software version that has been installed contains one of the following (variant 1 or variant 2), then the function can be used by making an entry in ...\user\paramtm.ini as described below. Otherwise the default setting applies.

Version 1:

```
[TMMODES]
...
; The inch/metric setting is considered for the code carrier
; -1 = inch/metric is ignored (default).
; Data transfer between the code carrier and the NCK/MMC takes place
; without consideration of the inch/metric setting. Behavior as up
to now.
; 0=mm
; It is assumed that storage of the data on the code carrier
; is to be executed in the unit mm
. If "inch" is set in the NCK
; all softkeys are disabled and code carrier
; functions start.
;1=inch
; It is assumed that the relevant data has been or is to be
; stored on the code carrier in inches.
. If "mm" is set in the NCK
; all softkeys are disabled and code carrier
; functions start.
DATABASE_LENGTH_UNIT=-1
```

Version 2:

```
[TMMODES]
...
; The inch/metric setting is considered for the code carrier
; !! NOTICE: The settings described here are only
; relevant if the machine data
; $MN_CONVERT_SCALING_SYSTEM=1 is set in the NC.
; If it is an old NC without an inch/metric
; conversion function, or if
; $MN_CONVERT_SCALING_SYSTEM=0, then the NC operates without
; inch/metric conversion. For this reason,
; no conversions are carried out in relation to the code carrier
either.
; -1 = inch/metric is ignored (default).
; Data transfer between the code carrier and the NCK/MMC takes place
; without consideration of the inch/metric setting. Behavior as up
to now.
; 0=mm.
; It is assumed that the relevant data has been or is to be
; stored on the code carrier in mm.
. If "inch" is set in the NCK
; all softkeys are disabled and code carrier
; functions start.
;1=inch.
; It is assumed that the relevant data has been or is to be
; stored on the code carrier in inches.
. If "mm" is set in the NCK
; all softkeys are disabled and code carrier
; functions start.
CODECARRIER_LENGTH_UNIT=-1
```

Setting for tool status

If the paramtm.ini or paramtm.txt of the software version that has been installed contains one of the following entries, then the function can be used by making an entry in ...\user\paramtm.ini as described below. Otherwise the default setting applies.

```
; Tool status: If a tool is removed from the NCK and transferred
; to an "external" medium (tool cabinet,
; code carrier, SINCOM), then you can use the following screens
; to specify which tool status bits should be
; saved.
; Code carrier: Since the standard wkonvert.txt file has entered 1
byte
; for the tool status and a max. of 92 have been written to the code
carrier
; up to now, CODECARRIER_TOOLSTATE_MASK receives the
; default value 92.
; If the value for CODECARRIER_TOOLSTATE_MASK is expanded,
; then the size of the dialog variables T9 in wkonvert.txt
; has to be adapted accordingly.


; 1=Active Tool
; 2=Allowed
; 4=Disabled
; 8=Measured
; 16=Warning limit reached
; 32=In change
; 64=Fixed place coding
; 128=Was used
; 256=Tool in buffer
; 512=Disabled, ignored (because of PLC)
; 1024=Out (unload)
; 2048=in (Load)
; 4096=Regular tool (permanent in NCK)
; 8192=
; 16384=
; Default is 4828 (4+8+16+64+128+512+4096),
; Default for codecarrier 92 (4+8+16+64)
...
CODECARRIER_TOOLSTATE_MASK=92
```

6.7.2 Structure of description file

Structure of description file

All data on the code carrier are stored in a particular order. This is defined during commissioning of the code carrier system. A conversion rule in the form of a description file is provided so that the tool management can read or write this data flow. This description file consists of correctly defined tool and cutting edge dialog data. Only this dialog data can actually be processed by the tool management. All the other data on the code carrier must not be assigned to any dialog variables as otherwise it will not be processed. An OEM application would, however, also be able to access this data.

The description file can be created as an ASCII file using a standard editor. The file name must be entered in mmc.ini with **WToolIdSysKonv = wkonvert.txt**.

Prerequisite

Minimum requirements for wkonvert.txt:

- Name
- Location type
- Subtype
- Number of cutting edges (if cutting edge data available)
- Tool size (the part of the size can be omitted which is hidden via paramtm.ini *)

  SHOW_TOOLSIZE_ONLY_LEFT_RIGHT=0

  SHOW_TOOLSIZE_COMPONENTS=left=True, right=True, top=True, bottom=True

Tool dialog data

The tool dialog data is defined as follows:

Dialog variable	Data type	Designation	Assignment $TC...
T1	String	Tool name, max. 32 characters	$TC_TP2
T2	Integer	Duplo number	$TC_TP1
T3	Integer	Number of cutting edges	$P_TOOLND[tnr] tnr=tool number
T4	Integer	Tool size left in half locations	$TC_TP3
T5	Integer	Tool size right in half locations	$TC_TP4
T6	Integer	Tool size upper in half locations	$TC_TP5
T7	Integer	Tool size down in half locations	$TC_TP6
T8	String	Magazine location type	$TC_TP7*
T9	Integer	Tool status	$TC_TP8
T10	Integer	Type of tool monitoring	$TC_TP9
T11	Integer	Type of tool search	$TC_TP10
T12	Integer	Magazine location type Previously, only the name of the magazine location type could be stored as a string via dialog variable T8. The assignment between location type number and location type name and vice versa is made in the tool management database.	$TC_TP7

* The character string which is stored there is an HMI internal location type which is assigned the value in $TC_TP7. This text is defined via the tool management start-up and stored in the database.

Note

If chips are to be exchanged between several machines, the following rule applies if T12 is used.

The location type names must be present in the same sequence on all these machines (with the same location type numbers).

If changes are made to the conversion file, old code carriers can no longer be read!

Code carrier chip/SINTDC:

If the user has defined the dialog variable T11 in the conversion file wkonvert.txt for "Type of tool search, $TC_TP10", the value from the NCK is written to the chip and written back to the NCK at read in. If T11 is missing, the value is 0 in $TC_TP10 after the chip is read in.

Cutting edge dialog data

The cutting edge dialog data is defined as follows:

Dialog variable	Data type	Designation	Assignment $TC...
C1	Integer	Subtype	$TC_DP1
C4	Integer	Length of cutting edge	$TC_DP2
		Geometry tool length compensation	
C5	Double	Length 1	$TC_DP3
C6	Double	Length 2	$TC_DP4
C7	Double	Length 3	$TC_DP5
		Geometry tool radius compensation	
C8	Double	Length 1	$TC_DP8
C9	Double	Length 2	$TC_DP9
C10	Double	Radius 1	$TC_DP6
C11	Double	Radius 2	$TC_DP7
C12	Double	Angle 1	$TC_DP10
C13	Double	Angle 2	$TC_DP11
		Wear tool length compensation	
C14	Double	Length 1	$TC_DP12
C15	Double	Length 2	$TC_DP13
C16	Double	Length 3	$TC_DP14
		Wear tool radius compensation	
C17	Double	Length 1	$TC_DP17
C18	Double	Length 2	$TC_DP18
C19	Double	Radius 1	$TC_DP15
C20	Double	Radius 2	$TC_DP16
C21	Double	Angle 1	$TC_DP19
C22	Double	Angle 2	$TC_DP20
		Base dimension/adapter dimension tool length compensation	
C23	Double	Basic length 1	$TC_DP21
C24	Double	Basic length 2	$TC_DP22
C25	Double	Basic length 3	$TC_DP23
C26	Double	Undercut angle	$TC_DP24
C27	Integer	Reverse insert	$TC_DP25
C28		Cutting edge number for addressing variables	-
C29 *	Integer	Downtime in minutes	$TC_MOP2
C30 *	Integer	Prewarning limit for tool life in minutes	$TC_MOP1
C31	Integer	Number of pieces still to be produced	$TC_MOP4
C32	Integer	Prewarning limit for number of pieces still to be produced	$TC_MOP3
C33	Double	Set downtime in minutes	$TC_MOP11
C34	Integer	Unit quantity setpoint	$TC_MOP13

Dialog variable	Data type	Designation	Assignment $TC...
C35	Double	Prewarning limit for wear	$TC_MOP5
C36	Double	Wear	$TC_MOP6
C37	Double	Setpoint wear	$TC_MOP15
C38 *	Double	Downtime in minutes	$TC_MOP2
C39 *	Double	Prewarning limit for tool life in minutes	$TC_MOP1
C40 *	Double	Quantity	$TC_MOP4
C41	Double	Pre-warning limit for count	$TC_MOP3
C42	Double	Unit quantity setpoint	$TC_MOP13

The dialog variables C2 and C3 are managed only internally.

Note

C38 and C39 can only be used as alternatives for C29 and C30. C40, C41 and C42 can only be used as alternatives for C31, C32 and C34.

User tool parameters and the new monitoring parameters are now used for code carriers. The following new dialog variables are available for the file "wkonvert.txt":

A1 - A10: User tool data (see $TC_TPCx[t])

U1 - U10: User cutting edge data (see $TC_DPCx[t,d])

S1 - S10: User monitoring data (see $TC_MOPCx[t,d])

Data type "Double" is defined for the dialog variables A, U and S.

Data types

The following data types are defined for dialog variables:

- Integer: Value range -32768 to 32767
- Double: Double-precision floating point
- String: String made up of ASCII characters

Keywords

The assignment of code-carrier data to dialog data is made using the code-carrier description file. The description file can be created and edited as an ASCII file using a standard editor. The code-carrier files is structured as lines whereby each line is prefixed by one of the following **keywords**:

Inverted comma

The **'** (single quotation mark) marks the beginning of a comment. The characters that follow are skipped.

Example: ' This is a comment. ¶

Note

This format for the beginning of a comment is used only in the description file for code carriers. Otherwise, the beginning of a comment is introduced by a semicolon (;).

Datalen

DATALEN=CONST | VARIABLE 0x<delimiter>

The following data have a constant (**CONST**) or a variable (**VARIABLE**) data length. Data with variable length are terminated with 0x<delimiter>.

Example:

```
DATALEN=VARIABLE 0x0A ' variable data length, delimiter LF
```

DEFINE_KEYWORD

DEFINE_KEYWORD=<keyword> <value><keyword> := any user keyword for indicating a new data section on the code carrier

<value> := "<string>" or 0x<hexvalue>

Definition of the keyword <keyword> with the value <value>

Example:

```
DEFINE_KEYWORD=DATA _OEM "OEM"
DEFINE_KEYWORD=DATA_SIN840D 0x840D
```

<keyword>

A keyword defined by **DEFINE_KEYWORD** that identifies a new data section on the code carrier. The item following **<keyword>** in the code carrier description file must contain the value <value> defined with **DEFINE_KEYWORD**.

Item

Item<n>**=<line>**

<n> := Consecutive number of code carrier data, ascending from 1 without gaps

<line> := <(max.) length in bytes> <code carrier data format> <dialog variable>

<Code carrier data format> :

<dialog variable> : Assignment of code carrier to dialog datum

If a user keyword is defined immediately before Item<n>, <dialog variable> takes the value <keyword>

Conversion rule for code carrier data <n>

Example:

```
 Item1   32   ASCII   T3   '   Convert tool identifier to/from
                           '   Tool dialog data 3
```

BItem

BItem<n>**=<**line**>**

<n> := consecutive number of code carrier data within block<i>, in ascending order from 1 without gaps

<line> := analog Item<n>

Conversion rule for code carrier data <n> within a block. If the code carrier data is assigned a tool dialog data T<n>, the dialog data is assigned the first value of the code carrier data in the block.

Example:

```
BItem1  1    BCD  C1, T2  '   Convert subtype before/after cutting
                          '   edge
                          '   dialog data 1 and tool dialog data 2
                              (1st value of block is relevant for T2)
```

Block and End_Block

Block<n> <repetition rule>

<n> := Consecutive number of block, ascending from 1 without gaps

<repetition rule> := * Item<n> **| CONTIGUOUS** BItem1

This is followed by a block of data **BItem**<n> (up to to keyword **End_Block**<n>), which is stored on the code carrier in accordance with the <repetition rule>.

Note

For Block<n> *Item<n>, Item<n> must be defined before Block<n>.

End_Block: End identifier for a data block defined with **Block**<n>

Example:

```
Block1 * Item6             ' Repeat Block1 corresponding to the
                             value of
                             Item6
Block1 CONTIGUOUS BItem1   ' Continue reading Block 1 until the
                           ' count variable
                           ' BItem1 stops returning a value
                           ' incremented by 1.
                           ' Write Block1 for the number of times
                             specified in the dialog variable
                             assigned to BItem1.
End_Block<n>
```

6.7.3 Data conversion rule

Code carrier data formats

The following code-carrier data formats are supported:

- Data format: Explanation
- ASCII: ASCII character set
- INT: 16-bit integer (Intel format)

  Range of values: -32768 <= INT <= +32767
- BCD: Binary-coded decimal number (if necessary, with sign and decimal point)

  Non-relevant decades are preassigned the value 0, left-justified.

  (Compare <code carrier data format> for **Item** / **BItem**)

Assignment between code carrier data and dialog data

Among other things, the conversion rule for **Item<n>** or **BItem<n>** includes the assignment to no/one/multiple dialog variables, where appropriate with a conversion rule, which is described in more detail in this section.

The general conversion rule for **Item<n>** or **BItem<n>** is as follows:

(B)Item<n>**=<**line**>**

```
<n>     :=  consecutive number of code carrier data,
            ascending without gaps
<line   :=  <(max.) length in bytes><code carrier data format>
>           <dialog variable>
```

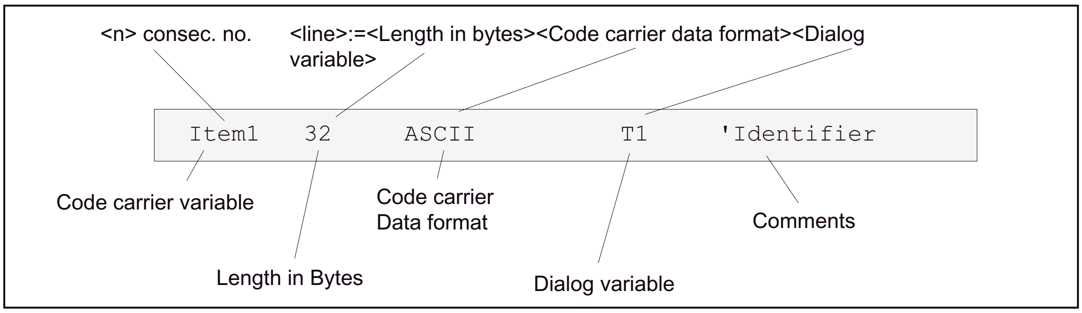


Figure 6-15 Conversion specification

Dialog variable

<dialog variable> := <dvar1>[=(<uv>)] [, <dvar2>[=(<uv>)] [,<dvar3>

[&<dvar4>]=(<uv>)] [, <dvarN>[=(<uv>)]

<dvar> := T<index> | C<index> | -

T = tool data,

C = cutting edge data,

index = index within tool/cutting edges dialog data

- = no assignment to a dialog variable

<dvar1>&<dvar2>=<uv> : Conversion rule applies to <dvar1> and <dvar2>

uv := <arithm. Op1> [<arithm. Op2>] .. [<arithm. OpN>]

arithm. Op := +<const> **|** -<const> **|** *<const> **|** /<const>

Example:

```
T2=(*10), T3=(/100 + 10)
```

or

uv := <replacement1> [<replacement2>] .. [<replacementN>]

replacement := <const1> [, <const2>] ..

[, <constN>] **^** <constM>

or

<const1..const2> ^ <const3>

const1 = lower limit value,

const2 = upper limit value

Note

When converting the dialog variable to the code carrier variable on writing, if there are several operands, the right operand is converted to the first left operand.

Example:

```
T2=(20..29 ^ 120 40,50 ^ 130)
```

The code carrier variable with the value 25 is converted to dialog variable T2 with the value 120 (read). Dialog variable T2 with value 120 is converted to code carrier variable with the value 20 (write).

or

uv := <Tetn>

Tetn := nth tetrad in byte sequence

Byte1, = Tet1 and Tet2

Byte2, = Tet3 and Tet4

Allocation of the tetrads of code carrier variables (in BCD format) to dialog variables.

Example:

```
T5=(Tet1), T6=(Tet2), T7=(Tet3), T8=(Tet4)
```

If the code carrier variable has the value 0x1234 for example, dialog variable T5 is assigned the value 1, dialog variable T8 the value 4.

or

uv := <compare>

compare := **<** <const> [INVSIGN] **|** **<=** <const> **|** **=** <const> **|** **>**

<const> **>=** <const>

Assignment of code carrier variable to a dialog variable according to the comparison result

INVSIGN

While reading: Sign for dialog variables.

While writing: Invert sign for code carrier variables.

Example:

```
C1=(<0 INVSIGN) , C2=(>=0)
```

- Read:

  A negative code carrier variable value corresponds to dialog variable C1, a positive value to dialog variable C2; dialog variable C1 is converted to a positive value.
- Write:

  Dialog variable C1 is multiplied by (-1). If the value is less than 0, then the code carrier variable is given the value from C1, otherwise the value from C2.

Note

Conversion specifications are only evaluated for dialog variables of data type "integer".

6.7.4 Example: Description file

Example of a description file or conversion file

The file name must be entered in ...\user\mmc.ini bei WToolIdSysKonv = **wkonvert.txt.**

The name of the file is, for example, wkonvert.txt:

Code carrier variable	Length (bytes)	Data format	Dialog variable	Comments
Item1	32	ASCII	T1	' Identifier
Item2	3	BCD	T2	' Duplo
Item3	2	BCD	T4 = (Tet1), T5 = (Tet2), T6 = (Tet3), T7 = (Tet4)	
'Tool size Left, right, top, bottom				
Item4	32	ASCII	T8	' Location type
Item5	1	BCD	T9	' Status
Item6	1	BCD	T3	' No. Tool noses
Item7	1	BCD	T10	' Type of tool monitoring
Item8	1	BCD	T11	' Type of tool search
' User tool data				
Item9	4	BCD	A1	' Tool OEM1
Item10	4	BCD	A2	' Tool OEM2
'Cutting edge data Block1 * Item6				
Bitem1	2	BCD	C1	' Subtype, type

Code carrier variable	Length (bytes)	Data format	Dialog variable	Comments
Bitem2	1	BCD	C4	' Cutting edge position
' Tool length compensation				
Bitem3	4	BCD	C5	' Length 1
BItem4	4	BCD	C6	' Length 2
BItem5	4	BCD	C7	' Length 3
'Tool radius compensation				
BItem6	4	BCD	C8	' Length 1
BItem7	4	BCD	C9	' Length 2
BItem8	4	BCD	C10	' Radius 1
BItem9	4	BCD	C11	' Radius 2
BItem10	4	BCD	C12	' Angle 1
BItem11	4	BCD	C13	' Angle 2
' Wear length compensation				
BItem12	4	BCD	C14	' Length 1
BItem13	4	BCD	C15	' Length 2
BItem14	4	BCD	C16	' Length3
'Wear radius compensation				
BItem15	4	BCD	C17	' Length 1
BItem16	4	BCD	C18	' Length 2
BItem17	4	BCD	C19	' Radius 1
BItem18	4	BCD	C20	' Radius 2
BItem19	4	BCD	C21	' Angle 1
BItem20	4	BCD	C22	' Angle 2
'Basic dimension length compensation				
BItem21	4	BCD	C23	' Basic length 1
BItem22	4	BCD	C24	' Basic length 2
BItem23	4	BCD	C25	' Basic length 3
BItem24	4	BCD	C26	' Tool clearance angle
BItem25	1	BCD	C27	' Reverse insert
BItem26	2	BCD	C29	' Downtime in minutes
BItem27	2	BCD	C30	' Prewarning limit for tool life
BItem28	2	BCD	C31	' Workpiece count
BItem29	2	BCD	C32	' Prewarning limit for tool life
' User cutting edge data				
BItem30	4	BCD	U1	' Cutting edge OEM1
BItem31	4	BCD	U2	' Cutting edge OEM2
'User monitoring data				
BItem32	4	BCD	S1	' Monitoring OEM1
BItem10	4	BCD	S2	' Monitoring OEM2

6.7.5 Example: Data string

Structure of a data string

conversion file: wkonvert.txt

Code carrier variable	Length in Bytes	Format	Dialog variable
Item1	10	ASCII	T1 Identifier, $TC_TP2
Item2	2	BCD	T2 Duplo, $TC_TP1
Item3	2	BCD	T4 = (Tet1), T5 = (Tet2), T6 = (Tet3), T7 = (Tet4) Tool size: left, right, top, bottom, $TC_TP3, 4, 5, 6
Item4	10	ASCII	T8 Location type, text for $TC_TP7
Item5	1	BCD	T3 Number of cutting edges, $P_TOOLND[tnr], tnr = tool number
Item6	4	BCD	A1 Tool OEM1, $TC_TPC1
Item7	4	BCD	A2 Tool OEM2, $TC_TPC2
Item8	2	BCD	C1 Subtype, type, $TC_DP1
Item9	4	BCD	C5 Geometry length1, $TC_DP3
Item10	4	BCD	C10 Geometry radius1, $TC_DP6
Item11	4	BCD	C14 Wear length 1, $TC_DP12

This conversion file can be used to read in the following data string or conversely to generate this string when writing:

```
626F687265725F31202000011116E6F726D616C2020202001D00010E3D000000502
05B00002E3B0000003B000E4562F2F
```

If working with SINTDC, this string corresponds to the interface between SINTDC and HMI Advanced.

Description

Dividing this string into individual values:

String	Value	Entry wkonvert.txt	Data types for dialog variables
626F687265725F312020	"Drill_1"	10 ASCII T1	T1 String Identifier, $TC_TP2
0001	1	2 BCD T2	T2 Integer Duplo, $TC_TP1

1111	1,1,1,1	2 BCD T4=(Tet1),
		T5=(Tet2),
		T6=(Tet3),
		T7=(Tet4))
			T4 Integer Tool size: Left
			T5 Integer Tool size: Right
			T6 Integer Tool size: Top
			T7 Integer Tool size: bottom

6E6F726D616C20202020	"Normal"	10 ASCII T8	T8 String Location type, text for $TC_TP7

String	Value	Entry wkonvert.txt	Data types for dialog variables
`01`	1	1 BCD T3	T3 Integer Number of cutting edges, $P_TOOLND[tnr] tnr=tool number
`D00010E3`	-10.300	4 BCD A1	A1 Double Tool OEM1, TC_TPC1
`D0000005`	-5	4 BCD A2	A2 Double Tool OEM2, TC_TPC2
`0205`	205	2 BCD C1	C1 Integer Subtype, $TC_DP1
`B00002E3`	2.3	4 BCD C5	C5 Double Geometry length1, $TC_DP
`B0000003`	3	4 BCD C10	C10 Double Geometry radius1, $TC_DP6
`B000E456`	0.456	4 BCD C14	C14 Double Wear length1, $TC_DP12
`2F2F`	End identifier (according to TIS_EOT=0x2F2F, mmc.ini)		

List of the INI Files

A

NOTICE
This chapter provides an alphabetical listing of all INI files indicating the sections that can be changed by the machine manufacturer or end user. **All other sections must not be changed!**

See also:

ACTLOG.INI (Page 274)

AEDITOR.INI (Page 275)

DINO.INI (Page 277)

DG.INI (Page 278)

DGOVW.INI (Page 279)

DH.INI (Page 280)

DPDH.INI (Page 281)

DPSIM.INI (Page 282)

HEADER.INI (Page 283)

HMIDESK.INI (Page 287)

IB.INI (Page 288)

IF.INI (Page 290)

KEYS.INI (Page 291)

LOGDRIVE.INI (Page 292)

MACHINE.INI (Page 295)

MBDDE.INI (Page 298)

MMC.INI (Page 303)

NETNAMES.INI (Page 308)

OEMFRAME.INI (Page 312)

PARAM.INI (Page 313)

PARAMTM.INI (Page 314)

REGIE.INI (Page 315)

SEDITOR.INI (Page 317)

SIMTOGEO.INI (Page 324)

TASKCONF.INI (Page 327)

A.1 ACTLOG.INI

Overview

The settings from the screen "Action log settings" are combined and from this a configuration file ACTLOG.INI generated.

Note

We recommended that the action log is configured using the setting dialog boxes and not using this file.

Setting the spindle number

When logging the program status, as standard, the spindle speeds of the first spindle are recorded. Other spindles can also be set in the ACTLOG-INI.

The entry IpoSpindleNum=Spindel Index must be changed to to the required Spindelindex in the section [ActionLogSettings]:

```
[ActionLogSettings]
ActionLogPath=F:\AC_LOG\crash.com
ActionLogSize=5000000
ActionLogBufSize=65536
AlarmLogPath=F:\AC_LOG\crash.com
AlarmLogSize=5000000
AlarmLogBufSize=4096
CrashLogPath=F:\AC_LOG\crash.com
CrashLogSize=5000000
DomainBufSize=20
PLCAlarms=3000
TripIds=0FFF38000000000000000000000000000000000000000001F000000000000
WriteInterval=-1
IpoTrace=1
IpoChannelNum=1
IpoSpindleNum=3
```

A.2 AEDITOR.INI

Overview

Information about the ASCII editor can be found in the AEDITOR.INI file.

The data regarding the size of the files that can be edited is in the section [MEMORY].

```
[MEMORY]
; default factor for maximum size of file
; valid values are: from 1 to 31
; factor x (1<=x<=31) means: you can edit a file up to
; a maximum size of x * 8 MB
; for example: factor 1 = maximum size of 8 MB, 7 = 56 MB
; attention: a greater factor needs more RAM
; the application can override this default factor
MemFactor=2
; default factor for maximum size of buffer file for editing NC
domains
; see explanation for MemFactor above
; the factor should be great enough for the whole NCU-memory
MemNCFactor=2
; number of undo steps, 0 = undo/redo not active, default = 0
(optional)
NrOfUndoSteps=10
; handling of undo / redo (optional)
; 0: Automatic, 1: Notify, 2: NoUndo, default = 0
UndoHandling=0
```

With this mode, it takes longer to prepare the NC file for processing in the HMI-Advanced.

```
; enable undo / redo for NC-files (=1) or disable (=0)
EnableUndoOnNc=0
```

Editor access authorizations are influenced in the section [ACCESSLEVEL]:

```
[ACCESSLEVEL]
GlobalReplace = 6
SettingHiddenCheck=7
ENABLE_SIMULATION=7
```

Parameters in the section [STEPS]

```
[STEPS]
; Mode for handling return key in step area
; 0: processing return key
; 1: process return key only at begin or end of a step or
; within a writable block of the step
; (same as 0 if HiddenCheck is active)
; 2: process return key only at begin or end of a step
NewLineInStepMode=0
; maximum number of blocks to search for a step, 0 == no limit
MaxNrOfBlocks=1500
```

```
; default mode for checking of readonly blocks, 0: no check,
; else check for readonly
ReadOnlyCheck=1
HiddenCheck=1
; default text for identification of readonly blocks
TextForReadOnlyCheck=;*RO*
TextForHiddenCheck=;*HD*

[MMC_Wizard]
Aeditor=Aeditor.com
```

A new line is inserted by pressing the <Return> key:

NewLineInStepMode

=0	**(Default)** If HiddenCheck=TRUE, whereby certain lines are hidden, <Return> can only be used at the end of the step (after the hidden lines). If HiddenCheck=FALSE, whereby all the lines are shown, <Return> can be used within a step.
=1	Although HiddenCheck=FALSE, <Return> can only be used at the beginning or end of a step (i.e. it cannot be used within a step).
=2	Irrespective of the value for HiddenCheck, <Return> can only be used at the beginning or end of a step and is rejected if used within a step.

A.3 DINO.INI

Access authorization

The access authorizations for series commissioning are assigned in the file DINO-INI and additional settings are defined for the archive:

```
[ACCESSLEVEL]
USER=3 ; read authorization for series commissioning
 ; customer password (standard)
PROGRAMMER=4 ; setup authorization for series commissioning
 ; corresponds to key-operated switch 3 (standard)

[Function]
; 0: ISO-formatted paper tape is not supported
; 1: ISO-formatted paper tape is supported
IsoActive=0
; 1: ask before overwriting CFG_RESET.INI, 0: don't ask
Ask_for_CFG_RESET.INI=0
; Directory to store ISO-formatted programs from
; ISO paper tape archive
; Not changeable by user interface (supported if IsoActive=1)
ISO-Path=/_N_WKS_DIR/_N_SHOPMILL_WPD
; writing ISO paper tape archive:
; 1: accept ISO programs only in ISO Path,
; 0: accept in all directories
; Not changeable by user interface (supported if IsoActive=1)
AcceptOnlyISO-Path=1
; 1: first unpack and then delete ZIP-files while reading
; a commissioning archive,
; 0: don't unpack, but store ZIP files in database
; not changeable by user interface
UnPack=1
; time to wait after a PLC memory reset in seconds
Wait for PLC=20
; reading internal pc-formatted archive that is not a setup archive:
; 0: destination of the unpacked files is determined through perhaps
; existing files with same name, if no such file exists destination
; is MMC
; else: destination is determined from the file header,
; i.e. from the location where the file was read from
TakeDestinationFromHeader=0

 [System]
; enable V24 setting by setting a non zero value
V24Settings=0
```

A.4 DG.INI

Overview

In the operator area "Diagnostics", the access to individual softkeys can be individually set by entering the protection level.

Enter the required protection level in the DG.INI file under [ACCESSLEVEL]:

```
[ACCESSLEVEL]
ALARMS=7               ; Alarms
SERVICE=7              ; Service displays
COMMUNIC-LOG=3         ; Communications log
PLC=3                  ; PLC status
PLC-CHANGE=2           ; PLC status: SK "Change Value" (change)
                       ; SK "Reset Value" (undo the change)
                       ; SK "Commit Value" (accept the change)
                       ; interlocked using the manufacturer's
                       password
REMOTE-DIAGNOSIS=4     ; Remote diagnostics
FILE=7                 ; File functions
RUN-LOG=7              ; Softkey "Action log" interlocked using the
                       user password
```

A.5 DGOVW.INI

Overview

This file is used for configuring additional customized signals in the "Diagnostics" → "Service Displays", "Service Overview" screen. Configured signals are added below the standard signals in this screen and can be viewed there (scrolling may be required). The file must be specifically created in one of the following directories: mmc2, addon, oem or user.

```
[GLOBAL]
; Number of additional signals
NrOfSignals=
; Bitmaps with customized icons to be displayed

[BMP]
100="<Bitmapname>"
101="..."
.
; One section per signal

[SIG1]
text=
Item=
expr=
.
 [SIG2]
.
[SIG<n>]
.
```

DGOVWTXT_xx.INI

The value "xx" in the file name contains the relevant language abbreviation. The file contains the language-specific texts for customized signals in the service overview. The file must be created explicitly in the mmc2, user, addon, oem, or …\language directory.

```
[TEXT]
; $T<Nr> = "Language-specific text"
; <Nr> stands for values from 1000 to 32767,
; Values below 1000 are reserved, e.g.:
$T1001="Signal s"
```

For details on configuring and for an example, refer to:

See also

Supplementing service displays on a user-specific basis (Page 36)

A.6 DH.INI

Overview

The access authorizations for directories of the data management are defined in the file DH.INI.

```
[ACCESSMASKS]
\wks.dir\*.wpd\*.mpf = 75775
\mpf.dir\*.mpf = 75775
\cus.dir\*.spf = 33773
```

A.7 DPDH.INI

Overview

Access authorizations for processing/editing workpiece programs are configured in the file DPDH.INI.

```
[ACCESSLEVEL]
PROGRAMMER=4
OPERATOR=5
ENABLE_DIALPROG=0
SetStandardWpSave=2
ENABLE_SIMULATION=7
```

A.8 DPSIM.INI

Overview

The run environment for the simulation is set in the file DPSIM.INI.

```
[ACCESSLEVEL]
;minimum level for changing setup standard:
SETUP=1
USER=3
OPERATOR=5

[MAIN]
 ;<>0: simulate program external modal (see also
 ; EXTERN_SIZE,EXTERN_CHECK)
EXTERN_MODAL=0
; simulate program external modal, if program
; size >= EXTERN_SIZE
; only if EXTERN_MODEL<>0
EXTERN_SIZE=150000
 ; only if EXTERN_MODEL<>0
EXTERN_CHECK=1
 ; <>0: reset tool data, if new workpiece is selected
 ; =0: use current tool data, if new workpiece is selected
SIM_LOAD_TOOLS=1

[PRELOAD]
;=1: TO_INI.INI and TO_ADDON.INI are preloaded by DPNCSIM.EXE
;=0: TO_INI.INI and TO_ADDON.INI are loaded by simulation
TOOLS=0
;= 1: all cycles are preloaded by DPNCSIM.EXE
; if section MAIN, entry CYCLE is set, additionally all cycles from
shadow directory are preloaded by DPNCSIM.EXE
;= 0: cycles are loaded by simulation, if cycle is called
CYCLES=1
;=2: load for simulation to \CUS.DIR, \CMA.DIR or \CST.DIR (default)
; load only cycles from NCK
; only evaluated if [MAIN]CYCLE is set and [PRELOAD]CYCLES=1
;=0: load all cycle (from NCK and MMC) for simulation to
 ; \CUS.DIR, \CMA.DIR or \CST.DIR
; only evaluated if [MAIN]CYCLE is set and [PRELOAD]CYCLES=1
;=1: use the cycle interface under \DP.DIR\SIM.DIR
CYCLEINTERFACE=2
```

A.9 HEADER.INI

Overview

User-specific icons and texts are configured in the HEADER.INI file that are output in the global cross-channel status display (header).

```
[CondStopIcons]
DefaultIcon=CondWait.bmp ;0 = No wait state

1=CondStop.bmp           ; No NC ready
2=CondStop.bmp           ; No mode group ready
3=CondStop.bmp           ; EMERGENCY STOP active
4=CondStop.bmp           ; Alarm with stop active
5=CondStop.bmp           ; M0/M1 active
6=CondStop.bmp           ; Block in SBL mode terminated
7=CondStop.bmp           ; NC stop active
8=CondStop.bmp           ; Read-in enable missing
9=CondStop.bmp           ; Feed enable missing
10=CondWait.bmp          ; Dwell time active
11=CondWait.bmp          ; Aux. function acknowledgment missing
12=CondStop.bmp          ; Axis enable missing
13=CondWait.bmp          ; Exact stop not reached
14=CondWait.bmp          ; Waiting for positioning axis
15=CondWait.bmp          ; Waiting for spindle
16=CondWait.bmp          ; Waiting for other channel
17=CondStop.bmp          ; Waiting for feedrate override
18=CondStop.bmp          ; Error in NC block
19=CondStop.bmp          ; Waiting for external NC blocks
20=CondWait.bmp          ; Waiting for synchronized action
21=CondWait.bmp          ; Block search active
22=CondStop.bmp          ; Spindle enable missing
23=CondStop.bmp          ;Axis feed override 0
24=CondWait.bmp          ; Waiting for tool change acknowledgment
25=CondWait.bmp          ; Gear step change
26=CondWait.bmp          ; Waiting for position control
27=CondWait.bmp          ; Waiting for thread cut
28=CondWait.bmp          ; Reserved
29=CondWait.bmp          ; Waiting for punching
30=CondWait.bmp          ; Waiting for safe operation
31=CondWait.bmp          ;No channel ready
32=CondStop.bmp          ; Oscillation active
```

```
33=CondWait.bmp         ; Axis exchange active
                        ; block change inhibited because axis exchange in progress
34=CondWait.bmp         ; Axis container rotation
35=CondWait.bmp         ; AXCT: Following axis active
36=CondWait.bmp         ; AXCT: Leading axis active
37=CondWait.bmp         ; AXCT: Follow up mode active
38=CondWait.bmp         ; AXCT: Internal status change
39=CondWait.bmp         ; AXCT: Axis/spindle inhibit
40=CondWait.bmp         ; AXCT: Corr. motion active
                        ; Axis container exchange: Overlaid movement
41=CondWait.bmp         ; AXCT: Axis exchange active
42=CondWait.bmp         ; AXCT: Axis interpolator active
43=CondWait.bmp         ; Waiting for compile cycle
44=CondWait.bmp         ; Access to system variable
45=CondStop.bmp         ; Search target found
                        ; (block search has found search target and the NCK has
                        stopped)
46=CondWait.bmp         ; Rapid retraction started
47=CondWait.bmp         ; AXCT: Waiting for spindle stop
                        ; Axis container exchange
48=CondWait.bmp         ; Machine data alignment (;New-Config)
49=CondWait.bmp         ; Axis exchange: Axis in coupling
50=CondWait.bmp         ; Axis exchange: Liftfast active
51=CondWait.bmp         ; Axis exchange: New-Config active
52=CondWait.bmp         ; Axis exchange: AXCTSW active
53=CondWait.bmp         ; Axis exchange: WAITP active
54=CondWait.bmp         ; Axis in another channel
55=CondWait.bmp         ; Axis exchange: Axis PLC axis
56=CondWait.bmp         ; Axis exchange: Axis oscillating axis
57=CondWait.bmp         ; Axis exchange: Axis JOG axis
58=CondWait.bmp         ; Axis exchange: Command axis
59=CondWait.bmp         ; Axis exchange: Axis OEM axis
60=CondWait.bmp         ; Coupled following axis
61=CondWait.bmp         ; Coupled-motion following axis
62=CondWait.bmp         ; Coupled slave axis
63=CondWait.bmp         ; Stop at cycle end M0
64=CondWait.bmp         ; Stop at cycle end M1
65=CondWait.bmp         ; Wait: Axis is at endstop
66=CondWait.bmp         ; "Master-slave switchover active"
67=CondWait.bmp         ;"Axis exchange: Axis single axis"
68=CondWait.bmp         ; "Stop: Target reached after block search"
```

```
69=CondWait.bmp          ; "Synchronism: Synchronous spindle"
70=CondWait.bmp          ; "Synchronous spindle deactivation position"
71=CondWait.bmp          ; "Waiting for enable of the transformer axis"
72=CondWait.bmp          ; "Waiting because of possible collision"
73=CondWait.bmp          ; "JOG: Position reached %2 %3 %4"
74=CondWait.bmp          ; "JOG: Direction inhibited %2 %3 %4"
75=CondWait.bmp          ; "Brake requirement %2 %3 %4"
76=CondWait.bmp          ; "Wait: G4 S%1 still: %2 U"
77=CondWait.bmp          : "Axial feed inhibit from PLC %2 %3 %4"
78=CondWait.bmp          ; "Waiting for axial feed enable"
79=CondWait.bmp          : "Axial feed inhibit from Synact %2 %3 %4"
80=CondWait.bmp          ; "Waiting for master spindle speed %1"
81=CondWait.bmp          ; "Waiting for parameter change %2 %3 %4"
82=CondWait.bmp          ; "Waiting for end of motion %2 %3 %4 before transformation
                         change"
```

```
[Pos1]
ChanStatFeedStopSpindleStop = 1 ;channel status, channel 1 is
displayed, superimposed
Spindles=1,2 ;from feed inhibit channel 1 and spindle inhibit
 ;for spindles 1 and 2

[Pos3]
SpindStat=1 ;spindle status of spindle 1
;ChanStatFeedStop = 3 ;channel status, channel 3 is displayed,
superimposed
 ;from feed inhibit, channel 3

[Pos5]
ChanStat = 2 ;channel status, channel 2 is displayed
;SpindStat= 3 ;spindle status, spindle 3 is displayed

[Pos9]
ChanStatFeedStopSpindleStop = 1 ;channel status, channel 1 is
displayed, superimposed
Spindles=1 ;from feed inhibit channel 1 and spindle inhibit
 ;of Spindle 1

[Pos11]
ChanStatFeedStop = 2 ;channel status, channel 2 is displayed,
superimposed
 ;from feed inhibit, channel 2

[Pos13]
SpindStat=2 ;spindle status of spindle 2
```

```
[Pos15]
ChanStatFeedStopSpindleStop = 1 ;channel status, channel 1 is
displayed, superimposed
Spindles=1 ;from feed inhibit channel 1 and spindle inhibit
 ;of Spindle 1


[UserIcons]
;UI_0 = <name.bmp> , <Position> ;name of the icon, position
...
;UI_31 = <name.bmp> , <Position> ;name of the icon, position
;USER_ICON_BASE = DBxx.DBByy
 ;number of the data block with the
 ;double word for user icon
 ;yy start byte of the double word


[UserTextInfo]
;USER_TEXT_BASE = DBxx.DBByy
 ;xx data block with interface for
 ;user messages in the header
 ;yy start byte for interface
;TextLength=33 ; activate text limiting, if
 ;user texts are used.


[OemIcons]
;Oem_ICON_BASE=DBxx.DBByy
;OI_0= = <name.bmp> , <Position> ;name of the icon, position
...
;OI_31 = <name.bmp> , <Position> ;name of the icon, position


[OEMTextInfo]
;OEM_NUMBER_TEXTFIELD=3 ;There are a max. of 3 text fields per line,
i.e.
 ;a total of 6 text fields in two lines are possible. ;The output
length is restricted to max. 38 characters
 ;per line.
;OEM_TEXT_BASE=DBxx.DBByy
```

A.10 HMIDESK.INI

Overview

Settings are only necessary in this file if HMI-Advanced is installed on a PG/PC.

```
[Global]
;Global configuration data
;definition of the desktop (minimum 2, maximum 4)
NumberOfDesktops = 2
;Control mode for using the mouse: Task bar
ControlMode = Icon
KeyboardControl = False

[Desktop1]
[Desktop2]
;Active desktop with task bar
Explorer = True
;
;Automatic start of an application after creating the desktop
; Start Regie
StartupApplication = "rngkrnl.exe"
;
;Desktop with customized color scheme
SetHMIColors = True
;
;Definition of start time
CreateDesktop = Immediately
;
;Keep user interface in foreground in window mode
WindowOnTop = False

[Desktop3]
[Desktop4].
```

A.11 IB.INI

Overview

Softkey access in the "Startup" operating area can be set by entering the protection level.

- Enter the required protection level in section [ACCESSLEVEL].
- The operator languages are defined in Section [LANGUAGE].

Note

In the [HELP_EXTENSION] section, the HelpFileExtension parameter must be set to 0 (WinHelp) if you want old, OEM-specific help files with the extension *.hlp to be displayed.

```
[ACCESSLEVEL]
RESET =2          ;NCK Reset                 (NCK Reset)
CHG_SCL_SYS=5     ;change scaling system     (Dimension system
                                             inch/metric)
MASCHDAT =4       ;Machine data              (Machine data)
DRIVE_CFG=2       ;Drive configuration       (Drive configuration)
BOOTFILE=2        ;Bootfile                  (Bootfile)
MOT_CTRL=2        ;Motor/valve selection     (Motor/valve selection)
VIEW=2            ;User views +              (User views + display
                  display options            options)
NC_ADDR=2         ;NCK Address               (NCK address)
SETUP_SWITCH=2    ;Start-up switch           (Commissioning switch)
PLC_STATE=3       ;PLC state                 (PLC status)
PLC_DATE=3        ;PLC date                  (PLC date)
PLC-CHANGE=2      ;PLC status: Interlocked by manufacturer
                  ;SK Change Value           (Change)
                  ;SK Reset Value)           (Undo change)
                  ;SK Commit Value           (Apply changes)
DRIVE_SERVO=2     ;Drives/Servo              (Drives/servo)
SET_COL=3         ;Colors                    (Colors)
SYST_SETTINGS=4   ;System settings           (System settings)
SET_LANG=2        ;Language                  (Languages).
SET_BTSS=2        ;Operator panel            (Operator panel front)
OPTION=2          ;Options                   (Options)
EDIT=1            ;Editor                    (Editor)
DOS=1             ;DOS shell                 (DOS shell)
TOOL_MNG=3        ;Tool management           (Tool management)
FILE=7            ;File functions            (File functions)
PASSWD=3          ;Set/Delete password       (Change/delete password)
CHG_PASSWD=7      ;SK "Password..."          (Softkey "Password...")
CHANGE_LANG=7     ;Change language           (Change language)
LICENSE=1         ;Set license               (Set license + NCK Reset)
SET_OPTIONS=1     ;Set options               (Change options + NCK Reset)
SHOW_LICENSE=4    ;Display license           (Show license)
SET_PD=2          ;Logic drives              (Logical drives)
```

```
[LANGUAGE]
;Softkey "select language" instead of softkey "change language"
; in the main menu of area setup.
;False: Softkey "change language" to toggle between the foreground and background
language.
;True: Softkey "select language" instead of softkey "change language"
;to select one of the installed languages. To use this add the following values
;to the entries of the LANGUAGE section in the mmc.ini:
;LanguageList= ... , GR
;FontList= ... , Europe
;FontListKO= ... , Europe2
;LBList= ... , German
SkSelectLngInsteadSkChangeLng=False


[Softkeys]
; Softkey "Advanced Reset..."  0 = not visible ; 1 = visible
AdvancedReset = 0

; LockResetIfPulsesEnabled = 0 → Reset is possible although pulses are enabled
; LockResetIfPulsesEnabled = 1 → Reset is not possible if pulses are enabled
LockResetIfPulsesEnabled = 0


[Reset]

ResetWith_p0972 = true
```

Meaning of the parameters

The parameters in the [SOFTKEYS] section have the following meaning:

- AdvancedReset:

  If the value is 1, the "Advanced Reset" softkey will be displayed in the Reset menu. This enables the NCK and drives to be reset separately. (Default setting: = 0)

- LockResetIfPulsesEnabled:

  If the value is 1 and the drives are in closed-loop control, a reset cannot be triggered. Enables must be removed manually.

- ResetWith_p0972:

  With Sinamics versions V2.5 SP1 and higher or V1.5 SP1 and higher, all control units are reset with p0972=3 during an NCK reset (hardware reset when cyclic communication fails). The same reset is also triggered after standard commissioning and factory settings have been loaded.

  The default setting is ResetWith_p0972 = true.

A.12 IF.INI

Overview

Functions to parameterize the help function and to output language-specific texts are available in the file IF.INI:

```
[TECHNOLOGY]
; "m" = milling
; "t" = turning
; "g" = grinding
; "s" = nibbling
; "p" = punching
; "a" = display all technologies
; Notation in the section e.g.:
Technology=m

[TEXTFILES]
; Siemens = Dh\Cst.dir\Ifs_gr.com
; SiemensI= Dh\Cst.dir\IfIT_gr.com turning ISO or
; SiemensI= Dh\Cst.dir\IfIM_gr.com milling ISO
; machinery construction OEM = Dh\Cst.dir\IfM_gr.com
; machinery construction OEM = Dh\Cst.dir\IfI_gr.com
; end user = Dh\Cst.dir\USER_gr.com
; Notation in the section e.g.:
Siemens=Cst.dir\ifs_gr.com
SiemensI=Cst.dir\ifit_gr.com
Manufacturer= …
End user=

[CONTEXT]
; = 0 context sensitivity not active
; = 1 context sensitivity active
; Notation in the section e.g.:
Extended_Sensivity=1

[TEXTSEARCH]
 ; = 1 text search only for instruction texts
 ; = 2 text search only for descriptive texts
 ; = 3 text search for instruction and descriptive texts
; Notation in the section e.g.:
Search type=3
```

A.13 KEYS.INI

Overview

Entries in this file allow the use of hardkeys of the operator panel front OP 010 and OP 010C or SINUMERIK keyboards with hotkey block and PLC softkeys.

Reference: Commissioning Manual "Supplementing the operator interface",
Chapter "Configuring OP hotkeys and PLC keys".

A.14 LOGDRIVE.INI

Overview

In the file LOGDRIVE.INI, logical drives are linked-in and the associated access authorizations assigned.

Note

The logical drives are set up via the HMI-Advanced operator interface under "Startup" → "HMI" → "Logic drives"; this enters them in the LOGDRIVE.INI file.

```
[PCU]
; device name of PCU for USB local
UsbFront=G:
; device name of PCU
UsbX203=H:
UsbX204=I:
; - <SEARCH PATH> : USER, OEM, ADD_ON, HMI_ADV, MMC2
; - it can be set up to 8 net devices or usb devices
; - for each device access rights can set for the applications
program, machine or services
; missing access rights are set to 7 (no limit) by default
; - all entries for net device are ignored if the software option
(Network drive management) is not set
; - all usb devices are ignored, if no tcu configuration exists.
 ACTTCU:\\FRONT always can be set
; all files LOGDRIVE.INI of the <SEARCH PATH> are read and the
contents are merged
; display machine data are append after the last entry of
LOGDRIVE.INI,
; but the maximum is limited to 8 net devices at all
; $MM_CMM_DIRECTORY_SOFTKEY_PATH1 (MD 9676) to
 $MM_CMM_DIRECTORY_SOFTKEY_PATH4 (MD 9679)
; the corresponding access level can be set by
; USER_CLASS_DIRECTORY1_P (MD 9510) to USER_CLASS_DIRECTORY4_P (MD
9513) for program and services
; USER_CLASS_DIRECTORY1_M (MD 9516) to USER_CLASS_DIRECTORY4_M (MD
9519) for machine
;
; ALL DISPLAY MACHINE DATA ARE IGNORED, IF A FILE LOGDRIVE.INI
EXISTS UNDER USER, OEM OR ADD_ON
; example for net device
; Connection<index>=\\server\share
; example for local device
; Connection<index>= X:
; example for usb device of current pcu/tcu, front
; one partition number may be specified for ALL tcu's. if omitted,
; the default partition is 1
: the partition number has no effect on a usb device of the pcu
; Connection<index>= ACTTCU:\\FRONT,1
; example for usb device of tcu TCU1, port X203. for second port use
X204
; a partition number may be specified. if omitted,
; the default partition is 1
: the partition number has no effect on a usb device of the pcu
; Connection<index>=TCU:\\TCU1:X203,1
; any directory path can be append (TCU:\\TCU1:X204,2\Directory1),
; but it must exist on the device

[Connections]
; Connection1 to Connection<N> are read with the corresponding
access
; levels, empty entries are ignored
; N=0 to 8, if missing ConnectionNum is set to 8
; set ConnectionNum to 0 to ignore all entries of LOGDRIVE.INI
; and display machine data
; ConnectionNum=8
; usb device of current pcu/tcu, front
Connection1=ACTTCU:\\FRONT,1
AccessProgram1=7
AccessMachine1=7
AccessServices1=7
```

Example: Entries in the logdrive.ini

```
[Connections]

Connection1=ACTTCU:\\FRONT
SK_Connection1=FRONT_USB
AccessMachine1=7
AccessProgram1=7
AccessServices1=7

.
.
.

Connection5=TCU:\\station2:X203
SK_Connection5=Labeling5
AccessMachine5=6
AccessProgram5=6
AccessServices5=6

Connection6=TCU:\\station3:X204
```

```
SK_Connection6=Labeling6
AccessMachine6=6
AccessProgram6=6
AccessServices6=6

Connection7=smb:\\<computer name>\<directory>
SK_Connection7=Network drive
Username7=Mustermann
Password7=431S0Y2E3K530I28
AccessMachine7=6
AccessProgram7=6
AccessServices7=6

Connection8=\\<computer name>\tmp
SK_Connection8=Drive%nlocal
AccessMachine8=5
AccessProgram8=5
AccessServices8=5
```

User name: (only for NW Windows)

```
Username<x> =
```

The user name can be entered in the user interface, e.g. Mustermann.

Password: (only for NW Windows)

```
Password<x> =
```

The password is displayed on the user interface with a *. It is encoded and entered in the logdrive.ini file.

A.15 MACHINE.INI

Overview

Access authorizations for the operating modes are set in the file MACHINE.INI in section [ACCESSLEVEL].

```
[ACCESSLEVEL]
;
; 7 = Key position 0 everyone in front of the machine,
; Suggestion: Permission only for NC-start, no selection of part-
program
; 6 = Key position 1 machine operator, production shop-floor
operators
; of production
; Suggestion: Permission to select part program, no editing
; 5 = Key position 2 machine setter (more experienced),
; the one to setup tools and test and run in new part programs
; Suggestion: Permission to modify (edit) part programs
; 4 = Key position 3 not yet used, same permissions as 5
; 1 = manufacturer of machine only, System specialist
; 0 = manufacturer of Numeric Control only (Siemens experts)
; at the state of delivery all users got (nearly) all permissions
for work


JOG_PRESET=7 ; Preset or Set actual value at mode JOG mode ==>5
Suggestion machine setter
JOG_HAND=7 ; Handweel at mode JOG
 ==>6 Suggestion operator
JOG_ANKRATZ=7 ; Scratch
 ==>6 Suggestion operator
JOG_INCREM=7 ; Set increment at mode JOG
 ==>6 Suggestion operator
JOG_SYNAKT=7 ; Synchronous actions at mode JOG
 ==>5 Suggestion machine setter
JOG_V_ISTW=7 ; watch windows of Actual-values,
 ==>7 Suggestion everyone
 ; Spindles, G functions, aux.-Functions,
 ; feed rate, tools or axis feed rate in JOG mode
JOG_V_ZOOM=7 ; Zoom actual values at mode JOG
 ==>7 Suggestion everyone
JOG_INCHMET=7 ; Switch inch/metric at mode JOG
 ==>6 Suggestion operator
REF_SISITEC=4 ; Commit safety integrated Ref.Point
 ==>4 key 3 to stay compatible
MDI_PROGBE=7 ; Program control at MDI mode
 ==>6 Suggestion operator
MDI_HAND=7 ; Handweel at MDI mode
```

```
 ==>6 Suggestion operator
MDI_EDITOR=7 ; Edit program at MDI mode
 ==>5 Suggestion machine setter
MDI_ED_DAT=7 ; Editor functions at MDI mode
 ==>5 Suggestion machine setter
MDI_ED_CLR=7 ; Clear buffer at MDI mode
 ==>5 Suggestion machine setter
MDI_V_ISTW=7 ; watch windows of actual values,
 ==>7 Suggestion everyone
 ; Spindles, G functions, aux. functions,
 ; Feed rate, tools or axis feed rate in MDI mode
MDI_V_ZOOM=7 ; Zoom actual values at MDI mode
 ==>7 Suggestion everyone
MDI_SYNAKT=7 ; synchronous actions at MDI mode
 ==>5 Suggestion machine setter
MDI_DATEI=4 ; file functions at MDI mode
MDI_INCHMET=7 ; Switch inch/metric at MDI mode
 ==>6 Suggestion operator
AUTO_OVERST=7 ; Overstore at AUTO mode
 ==>5 Suggestion machine setter
AUTO_DRF=7 ; DRF at AUTO mode
 ==>6 Suggestion operator
AUTO_PROGBE=7 ; Program control at AUTO mode
 ==>6 Suggestion operator
AUTO_SASUCH=7 ; Block search at AUTO mode
 ==>6 Suggestion operator
AUTO_HAND=7 ; Handweel at AUTO mode
 ==>6 Suggestion operator
AUTO_KORREK=7 ; Correct program at AUTO mode
 ==>5 Suggestion machine setter
AUTO_V_ISTW=7 ; watch windows of actual values,
 ==>7 Suggestion everyone
 ; Spindles, G functions, aux. functions,
 ; Feed rate, tools or axis feed rate in AUTO mode
AUTO_V_ZOOM=7 ; Zoom actual values in AUTO mode
 ==>7 Suggestion everyone
AUTO_V_NCPR=7 ; Program overview in AUTO mode
 ==>6 Suggestion operator
AUTO_OP_NCPR=7 ; load + select program in AUTO mode
 ==>6 Suggestion operator
AUTO_ED_NCPR=7 ; Alter enable status of Prog in AUTO mode
 ==>5 Suggestion machine setter
AUTO_SYNAKT=7 ; synchronous actions in AUTO mode
 ==>5 Suggestion machine setter
AUTO_INCHMET=7 ; Switch inch/metric in AUTO mode
 ==>6 Suggestion operator

[Customer]
MDASingleSTEP=yes
ONLY_MKS_ToolBaseDistToGo=no
NcVersionCheck=true
```

```
 ; Due to performance issues the functionality offered with respect
to
 ; block correction, block view and block search depends on the
 ; Nc-version currently used.
 ; However, to get access to the whole functionality, regardless of
the ; Nc-Version in use,
 ; set the flag NcVersionCheck to False.


[BlockSearch]
 ; Shows (SeruproEnabled=1) or
 ; hides (SeruproEnabled=0)
 ; the softkey used to do a block search in mode "programtest".
 ; The softkey appears in both dialogs
 ; "Blocksearch Searchposition" and "Blocksearch Searchpointer"
SeruproEnabled=1
 ; When doing a block search in mode "programtest" you can do
 ; a multi-channel block search. There are 4 channel-configurations
which
 ; define the NC-channels being affected by the block search:
 ; SeruproConfig =
 ; 1: selected/active channel
 ; 2: selected/active channel and all channels having selected the
same
 ; workpiece as the active channel
 ; 3: selected/active channel and all channels belonging to the
active
 ; channel's's mode group
 ; 4: all NC-channels
SeruproConfig=1


[SetZero]
 ; Display_Sign_Switch switch the sign of the tool length in the
scratch picture
 ; Display_Sign_Switch=0 mean - tool length
 ; Display_Sign_Switch=1 mean + tool length
Display_Sign_Switch=1
```

A.16 MBDDE.INI

Overview

The file MBDDE.INI is evaluated when booting. Settings are made to alarms and the actual log file in directory \mmc2.

```
[Alarms]
; determines the order in which the alarms are inserted in the alarm
list
; possible values:
; FIRST: makes newer alarms stand on top of the list
; LAST: inserts the newest alarm at the bottom
ORDER=LAST
; the alarm/message line is able to scroll through the alarms
; possible values:
; 0 or negative value: no scrolling
; 500 - 32767: interval in ms
RotationCycle=0
; determines whether or not the alarm server should connect to
ALARM_S/SQ
; possible values:
; 0 Alarm_S/SQ not handled
; 1 Alarm_S/SQ handled (default)
Alarm_S=1
; if Alarm_S/SQ handling is enabled this key determines if Quits
should
; be entered into the alarm log
; possible values:
; 0 Alarm_SQ quits are not entered into the alarm log
;(backward compatible)
; 1 Alarm_SQ quits are entered into the alarm log (default)
Protocol_Alarm_SQ=1
; If Alarm_S/SQ handling is enabled this key determines if the
; workaround regarding an error in the firmware of the PLC is
; active or not. This workaround corrects the listing of
; Alarm_S/SQ in the diagnosis but causes multiple entries in
; the log list of one alarm event (NEW, NULL, QUIT).
; Using a newer PLC firmware (PLC Besy Stand: S7 PLC_315 2DP
; 2AF03 12.30.03 or higher only necessary at NCU57x.1) this
; workaround can be deactivated to correct log writing.
;
; Attention: Setting the workaround inactive while using an PLC
; firmware from 12/30/2003 or before leads to unpredictable results!
;
; possible values:
; 0 Workaround active
; 1 Workaround active (default)
```

```
Alarm_S_FWA_Inactive=1
; path and file names of the alarm text lists
; The filename fragments must not exceed 6 char., because it is
extended
; automatically by the message module according to the selected
language.
; The alarm texts in the files specified under UserMMC, UserNCK ...
; overload the corresponding texts of the files defined in MMC, NCK
...
; Therefore the user should modify files according to the needs of
his
; system exclusively in UserMMC, UserNCK ...
; Creating alarm text with any DOS-editor may cause Problems with
special
; characters like ä, ö or ü. To nevertheless import DOS generated
files,
; add a blank and the string "DOS" after the name of the text file.
; Example: UserPLC=F:\dh\mb.dir\myplc_ DOS
; If you add or remove the parameter DOS, you have to touch
 (new date of file)
; the corresponding file (*.com) otherwise the change is not
accepted.


[TextFiles]
MMC=F:\MMC_52\dh\mb.dir\alm_
NCK=F:\MMC_52\dh\mb.dir\aln_
PLC=F:\MMC_52\dh\mb.dir\alp_
ZYK=F:\MMC_52\dh\mb.dir\alz_
CZYK=F:\MMC_52\dh\mb.dir\alc_
STANDARD_CYCLES=F:\MMC_52\dh\mb.dir\alsc_
SHOPMILL_MANUALTURN_CYCLES=
MEASURE_CYCLES=F:\MMC_52\dh\mb.dir\almc_
PLC_PMC=
USER_CYCLES=
UserMMC=
UserNCK=
UserPLC=
UserZYK=
UserCZYK=
UserSTANDARD_CYCLES=
UserSHOPMILL_MANUALTURN_CYCLES=
UserMEASURE_CYCLES=
UserUSER_CYCLES=
UserPLC_PMC=
; NCU-specific texts in M:N environments
; the entry names are build by concatenating the string "PLC"
 or "NCU" with
; the name of a NCU taken from netnames.ini file.
; Entry names are case sensitive!
; the entry values follow the same rules as in the Textfiles section
```

```
 of this file
; the commented examples below correspond to the netnames.bsp
 file delivered

[net TextFiles] (only for SINUMERIK powerline)
;PLCRechteNCU = f:\dh\mb.dir\rechts
;NCURechteNCU = f:\dh\mb.dir\rechts
;PLCMittlereNCU = f:\dh\mb.dir\mitte
;NCUMittlereNCU = f:\dh\mb.dir\mitte
;PLCLinkeNCU = f:\dh\mb.dir\links
;NCULinkeNCU = f:\dh\mb.dir\links
; drive and device type specific alarm text files

[DriveTextFiles]
ALDEV_80E5=f:\dh\mb.dir\aldvs_
ALDRV_80E5=f:\dh\mb.dir\aldrs_
;ALDEV_808F=f:\dh\mb.dir\aldvu_
;ALDRV_808F=f:\dh\mb.dir\aldru_
ALDEV_x=f:\dh\mb.dir\aldvp_
ALDRV_x=f:\dh\mb.dir\aldrp_
[IndexTextFiles]
ALNX=f:\dh\mb.dir\alnx_
ALSI=f:\dh\mb.dir\alsi_
; default definition of the priorities of the different alarm types

[DEFAULTPRIO]
CANCEL=100
RESET=100
POWERON=100
NCSTART=100
PLC=100
PLCMSG=100
MMC=100
; This section defines different characteristics of the log that
; specifies how particular error messages are to be registered.

[PROTOCOL]
```

Details on generating filter expressions to control the number of messages seen are provided in the Chapter "Setting the memory space for the alarm log".

```
; Criteria for selecting the alarm messages to be recorded. The
notation
; is described in SINUMERIK 840D OEM package HMI User's Manual.
Filter=Expression
; Characterizes the numbers of entries in the log file. If the
actual
; number exceeds this value, the server starts to override old
entries
```

```
; (principle: ring buffer)
Records=150
; Mode of alarm log writing
; if a number of less than -1 is specified, this number of log
 files is
; kept in parallel, and one of them is actually used
; if a number of -1 is specified, the log file is not
 automatically updated
; if 0 is specified, the log file is immediately updated
; if a number above 0 is specified, the log file is updated
 after the
; specified number of seconds elapsed with no intermediate alarm
state
; changes. The log file is always updated when you send the
; DDE command: "AlarmProt" to the MBDDE-server
; (principle: ring buffer)
---------------------------------------------------------------
DiskCare=-1 # attention: the mode changed from V4 to V5 !
The multi-file strategy can also be selected in the
Ini file using the entry DiskCare.
; information about the keys that may be used to clear an alarm


[KEYS]
; A function key (e.g. ESCAPE) is assigned the cancel function.
; This makes it possible to acknowledge alarm cancellations on the
; PC development system, too.
---------------------------------------------------------------
Cancel=Escape
; This section specifies the names and paths of the help files that
; contain help texts in WinHelp format.
; This list includes a maximum of 10 entries.
; The file name fragments must not exceed 6 characters, because it
is
; extended automatically by the message module according to the
; selected language. A file name extension may also be added
(optional).
; For additional information, see SINUMERIK 840D OEM package
; HMI User's Manual.


[Helpcontext]
File0=hlp\alarm_ .pdf
For OEM users:
; reserve some sender names of HMI alarms. The names mbdde,
; NCU and PLC are reserved for mbdde internal purposes.


[MmcAlarmDisable]
DisableSenderOfMmcAlarm0=mbdde
DisableSenderOfMmcAlarm1=NCU
```

```
DisableSenderOfMmcAlarm2=PLC
DisableSenderOfMmcAlarm3=
DisableSenderOfMmcAlarm4=
DisableSenderOfMmcAlarm5=
DisableSenderOfMmcAlarm6=
DisableSenderOfMmcAlarm7=
DisableSenderOfMmcAlarm8=
```

A.17 MMC.INI

Overview

Settings regarding the current system structure and the communication of the individual system components and the languages available at the HMI are configured in the file MMC-INI.

```
[GLOBAL]
; connectiontype 0 = default setting: Connection SINUMERIK 810D/840D
; 1 = Connection to SINUMERIK 840Di
connectiontype=0
NcddeServiceName=ncdde
ModeWithoutNc=False
ProductCode=MMC
; delay time for screensaver
MMCScreenOffTimeInMinutes=5
; ATTENTION PLEASE FOR USING a SIM1 - MACHINE !!!!!
; for working with a SIMNC arrange the following entries AND
; activate the [NCDDE_STARTCMD] section later in this mmc.ini
; ATTENTION PLEASE FOR USING a SIM1 - MACHINE !!!!!
; for using M:N function, set NcddeMachineName=MachineSwitch
; for working without NC, set NcddeMachineName=local
; for working with SIMNC, set NcddeMachineName=SIM1
; for connecting to an NC, set NcddeMachineName=NCU840D
NcddeMachineName=NCU840D
; for using M:N function, set NcddeDefaultMachineName=net:NCU_1
; for working without NC, set NcddeDefaultMachineName=local
; for working with SIMNC, set NcddeDefaultMachineName=SIM1
; for connecting to an NC, set NcddeDefaultMachineName=NCU840D
NcddeDefaultMachineName=NCU840D
; for using M:N function, set NcddeMachineNames=net,NCU840D
; for working without NC, set NcddeMachineNames=
; for working with SIMNC, set NcddeMachineNames=SIM1
; for connecting to a NC, set NcddeMachineNames=NCU840D
NcddeMachineNames=NCU840D
; for connecting to an SIM, add NcddeMachineNamesAdd1=SIM0
NcddeMachineNamesAdd1=SIM0,SIM1
; for using M:N function, set NcddeStartupFile=ncdde6.nsk
; for working without NC, set NcddeStartupFile=ncdde202.nsk
; for working with SIMNC, set NcddeStartupFile=sim1dde5.nsk
; for connecting to an NC, set NcddeStartupFile=ncdde6.nsk
NcddeStartupFile=ncdde6.nsk
; define a network-wide unique MMC name (8 characters) here
NcddeMmcName=__XXXX__
; 1 makes servers such as NCDDE, MBDDE, and DH servers visible.
; With ALT +TAB, they can be brought to the foreground to
; parameterize them, if necessary, for a trace (for service).
; Comment out this entry for this purpose:
; ServerVisible = 1

[Local] for OEM users
ADDRESS0=0,NAME=/NC,LINE=0
ADDRESS1=0,NAME=/PLC,LINE=0
ADDRESS4=/SELF,0
; Example: 840D via OPI connection

[NCU840D] only SINUMERIK powerline
ADDRESS0=13,NAME=/NC,SAP=d
ADDRESS1=13,NAME=/PLC,SAP=1
ADDRESS4=/SELF,10000103
; Example:
; WINCOS-->CP-->TCPIP-->CP-->LINCOS-->NC/PLC/DRIVE0-5
; 1.) change TCPIP-Address below to your target TCPIP-Address
; 2.) do not forget to start your windows CP (e.g. cp_840di.exe)
; 3.) note: no disconnection detection in some cases
;[NCU840D] only SINUMERIK solution line
;
ADDRESS0=192.168.214.1,LINE=10,NAME=/NC,SAP=040d,PROFILE=CLT1__CP_L4
_INT
;
ADDRESS1=192.168.214.1,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L
4_INT
;
ADDRESS2=192.168.214.1,LINE=10,NAME=/CP,SAP=0501,PROFILE=CLT1__CP_L4
_INT
; ADDRESS10=192.168.214.1,LINE=10,NAME=/DRIVE0,SAP=0201,SUBNET=004B-
00000008:3, PROFILE=CLT1__CP_L4_INT
;
ADDRESS11=192.168.214.1,LINE=10,NAME=/DRIVE1,SAP=0a00,PROFILE=CLT1__
CP_L4_INT
;
ADDRESS12=192.168.214.1,LINE=10,NAME=/DRIVE2,SAP=0b00,PROFILE=CLT1__
CP_L4_INT
;
ADDRESS13=192.168.214.1,LINE=10,NAME=/DRIVE3,SAP=0c00,PROFILE=CLT1__
CP_L4_INT
;
ADDRESS14=192.168.214.1,LINE=10,NAME=/DRIVE4,SAP=0d00,PROFILE=CLT1__
CP_L4_INT
;
ADDRESS15=192.168.214.1,LINE=10,NAME=/DRIVE5,SAP=0e00,PROFILE=CLT1__
CP_L4_INT
```

The language can also be selected via the HMI operator interface in the operator area "Startup" → "HMI" → "Languages":

```
[LANGUAGE]
Language=UK
LanguageFont=Europe
Language2=GR
LanguageFont2=Europe
```

```
LanguageList=GR, UK, SP, FR, IT, CH
FontList=Europe, Europe, Europe, Europe, Europe, China
FontListKO=Europe2, Europe2, Europe2, Europe2, Europe2, China
LBList=German, English, Spanish, French, Italian, Chinese
```

```
[CONTROL] only relevant for HMI-Advanced on PC/PG
; Resolution: 0=fixed (640x480 default), 1=variable
Resolution=0
; BaseScreen specifies the resolution, for which the applications
; were originally developed (typically: 640x480)
; BaseScreen - default: 640x480
BaseScreen=640x480
```

Note

Switching off motion keys

The CPF_Disabled entry is evaluated in the [CONTROL] section of the MMC.INI configuration file: If the value = 1, the CPF menu will not be displayed.

If a mouse is connected to the control, the default position for the cursor is the upper right corner of the screen. The entry below can be used to disable this behavior:

```
;MoveCursor=0: do not move cursor to upper right corner
;MoveCursor=0
[Services] only relevant for HMI-Advanced on PC/PG
FloppyDisk= a:
PCCard=
```

Note

If you want to adjust the font size for different display modes/languages, contact the Siemens hotline.

OPI settings consist of the display machine data, which can also be set on a technology-specific basis via the operator interface of the HMI in the operating area "Startup" → "Machine data":

```
[OPI settings]
$MM_LCD_CONTRAST=7
$MM_DISPLAY_TYPE = 1
$MM_DISPLAY_MODE = 2
$MM_FIRST_LANGUAGE = 1
$MM_DISPLAY_RESOLUTION=3
$MM_SPIND_DISPLAY_RESOLUTION=3
…
$MM_MD_DRV_TEA_FILTER = 0
```

```
$MM_MD_DRV_TEA_IDX_LIMIT = 0
$MM_MD_TEXT_SWITCH=0
$MM_ACTION_LOG_MODE = 0xFE
$MM_PROGRAM_SETTINGS = 0
$MM_SW_OPTIONS = 0
$MM_HMI_TESTAUTOMAT_OPTION = 0
$MM_HMI_WIZARD_OPTION = 0
$MM_TRACE=0
```

The action log can also be set via the operator interface of the HMI in the operating area "Startup" → "HMI" → "System settings":

```
; ActionLog (action log) parameters


[ActionLog]
; CommitLatency gives the maximum number of seconds that elapse
after creation of an ActionLog
; record is written to disk. If this value is less than 0 there is
no
; time controlled automatic writing to disk at all
; Default: CommitLatency = -1
; no automatic writing
CommitLatency = -1
; DirectoryPath specifies the directory of the log files
DirectoryPath = D:\AC_LOG
; FileSize gives the upper size limit of the log file, in bytes
FileSize = 40000
; VariableN allows monitoring of max 10 NC/PLC variables. The first
; parameter is numeric and provides the monitoring mode:
; if Bit0 is set and the value of the variable changes, the value
changes
; are logged in the ActionLog files
; if Bit1 is set and the value of the variable changes, the
ActionLog files
; are committed to disk
; Default: Variable0 = 1, /Channel/ProgramInfo/msg
; monitor part program messages
Variable0 = 1, /Channel/ProgramInfo/msg
; CommittedAlarms holds up to 10 comma separated alarm numbers.
; Changes of these alarms commit
; ActionLog data to disk
; Default: CommittedAlarms = 3000
; commit on emergency stop
CommittedAlarms = 3000
; for working with SIMNC, uncomment the following section descriptor
; this causes a sim-process to be started that will be used as a
machine


;[NCDDE_STARTCMD]
CMD=dpncsim.exe /T:1 /P:0
WND_CLASS=SIM_LOCAL_NCK
```

```
WND_NAME=SIM_LOCAL_NCK
STARTUP_TIMEOUT=50
;WAIT_SYNC_MESSAGE=
CLOSE_TIMEOUT=30

[WP_SELECT]
; for displaying a message that TOA/TMA-data on NCK will be
overwritten
;TOA_WARNING_MESSAGE = yes/no
TOA_WARNING_MESSAGE=yes

[PCU]
; device name of PCU for USB local
UsbFront=G: ' access rights of the PCU USB device (0-7) for the
application machine, program and services
AccessMachine=7
AccessProgram=7
AccessServices=7

[PROGRAM]
;0=message to user if a program of the DAT-file for MCSE was not
found
; or a channel doesn't exist (default)
;1=skip all entries of DAT-file if the program or a channel was not
found
IgnoreMissingProgramAndChan=0
;0=create a workpiece (optional by template) without opening by
editor
;1=open part program, job list or DAT-file after creating
; new workpiece (default)
OpenFileAfterNewWPD=1
```

The following is applicable to the **multi channel sequence editor option** in the USER directory:

```
[PROGRAM]
IgnoreMissingProgramAndChan=1
OpenFileAfterNewWPD=0
```

The "Channel assignment missing" message does not appear when workpieces are newly created using only a job list without MPF.

See also

SEDITOR.INI (Page 317)

A.18 NETNAMES.INI

Overview

In the m:n assignment in file NETNAMES.INI, you define which PCU operator components (HMI) can be connected with which NCU in the network (powerline) or which TCU can be switched to which PCU/NCU (solution line).

The following details are specified:

- Name of the operator component
- Name of the NCU
- Communication connections between the NCUs
- Unique assignments for booting

References: Function Manual, Expansion Functions: Several operator panel fronts on several NCUs, distributed systems (B3)

Proceed as follows

All configuration options are explained in file F:\mmc2\netnames.ini (powerline). It makes sense to maintain the following sequence when configuring NETNAMES.INI for m:n communication:

1. Copy the existing file F:\mmc2\netnames.ini to F:\user\netnames.ini.
2. Edit file F:\user\netnames.ini in an editor in the following steps:
   – Define groups (maximum of 24) in "logChanSetList"
   – Define the NCUs and the channels with unique names for the groups in "logChanList".
   – For the channels, define the NCU names in "logNCName", the channel number in the NCU in "ChanNum" and keep the softkey free yes/no (using the command IsAGap" = True/False).
   – For the HMI, specify the default machine "DEFAULT_logChanGrp" and the default channel in "DEFAULT_logChan" for booting.
3. Save file F:\user\netnames.ini.

Note

The NC writes the channel number of the HMI to DB19.DBB22.

The example below is simplified for 1:1 assignment of HMI : NCU so that conn_2 is eliminated and all channels are located on NCU_1.

Content of the netnames.ini file

```
; Detection part MMC identification
; PCU50 PL3 server

[own]
owner= PCU_1
```

```
; Description of possible connections
; Connection part

[conn PCU_1]
conn_1 = NCU_1
conn_2 = NCU_2
conn_3= NCU_3
EXTCALL_conns=conn_1, conn_2, conn_3

; Description of significant net parameters
; descriptive part, NCU components

[param network]
bus = opi

; MMC definitions

[param PCU_1]
mmc_typ = 0x40
mmc_bustyp = OPI
mmc_address = 1
mstt_address=3
name = PCU50
start_mode = ONLINE

[param NCU_1]
type =NCU_573
nck_address = 11
plc_address = 11
name = NCU1
PlcSymbolFile= ...

[param NCU_2]
type =NCU_573
nck_address = 12
plc_address = 12
name = NCU2
PlcSymbolFile= ...

[param NCU_3]
type= NCU_572
nck_address = 14
plc_address = 14
name = NCU3
```

```
PlcSymbolFile= ...

; Channel data

;****************

[chan PCU_1]
DEFAULT_logChanSet = Stat_1
DEFAULT_logChan = N1_K1
ShowChanMenu = True
logChanSetList = Stat_1, Stat_2, Stat_3

;***************************

[Stat_1]
logChanList = N1_C1, N1_C2

[N1_K1]
logNCName = NCU_1
ChanNum = 1

[N1_K2]
logNCName = NCU_1
ChanNum = 2

;***************************

[Stat_2]
logChanList = N2_C1, N2_C2

[N2_K1]
logNCName = NCU_2
ChanNum = 1

[N2_K2]
logNCName = NCU_2
ChanNum = 2


;***************************

[Stat_3]
logChanList = N3_C1, N3_C2
```

```
[N3_K1]
logNCName = NCU_3
ChanNum = 1

[N3_K2]
logNCName = NCU_3
ChanNum = 2


;End of file
```

See also

Configuration of a channel menu for direct channel selection (1:1 configuration) (Page 53)

Configuring a double-channel display (Page 56)

A.19 OEMFRAME.INI

Overview

The starting behavior of OEM-applications is configured in the file OEMFRAME.INI.

```
[SysMeter]
; let OEMFrame just look for Windows which have been
 ; created by the OEMApp
; default is 0 (means NO)
;fSearchOnlyForTaskWindow=1
; let OEMFrame delay INIT_COMPLETE, if set to -1,
 ; OEMFrame will not send any WM_INITCOMPLETE at all,
 ; this value must be given in microseconds
;nDelayInitComplete=10000
; switching WS_SYSMENU, WS_MINIMIZEBOX and WS_MAXIMIZEBOX off
WindowStyle_Off = 720896
; other possible task entries aren't used
WindowStyle_On =
x =
y =
Width =
Height =
```

A.20 PARAM.INI

Overview

The tool compensation data that the user enters via the HMI operator interface are input into this file.

```
[ACCESSLEVEL]
READ_SYSVAR=7
EDIT_VIEW=7
;Tool compensation user data
;Tool-specific parameters

[ToolParams]
UserDataParamName1=T_Param_1
UserDataParamName2=T_Param_2
UserDataParamName3=T_Param_3
UserDataParamName4=T_Param_4
UserDataParamName5=T_Param_5
UserDataParamName6=T_Param_6
UserDataParamName7=T_Param_7
UserDataParamName8=T_Param_8
UserDataParamName9=T_Param_9
UserDataParamName10=T_Param_10
; cutting edge-specific parameter

[ToolEdgeParams]
UserDataParamName1=D_Param_1
UserDataParamName2=D_Param_2
UserDataParamName3=D_Param_3
UserDataParamName4=D_Param_4
UserDataParamName5=D_Param_5
UserDataParamName6=D_Param_6
UserDataParamName7=D_Param_7
UserDataParamName8=D_Param_8
UserDataParamName9=D_Param_9
UserDataParamName10=D_Param_10
; Tool compensation data
```

A.21 PARAMTM.INI

Overview

All data that describe the operator interface of the tool manager are saved in the file ...\user\paramtm.ini.

See also

Structure of the paramtm.ini file (Page 187)

A.22 REGIE.INI

Overview

Settings to start and end the HMI application and the associated sub-systems are made in the file REGIE.INI.

```
[Miscellaneous]
; needed if regie is to support an 'EXIT' button
; set 'ExitButton' to True, and 'ExitButtonIndex'
; identifies index of exit button
; The softkey text for this button MUST be entered into
; \mmc2\language\re_xx.ini
; where xx is the language-abbreviation e.g. :
; re_gr.ini (German text)
; re_uk.ini (English text)
ExitButton = True
ExitbuttonIndex = 15
; Protection level for exit button
ExitButtonAccessLevel = 7
; Dialog box to ask user
ExitButtonQueryUser = True
; Notes on other entries can be found in the OEM documentation
; Activating the temperature monitoring for PCU50
; TemperatureMonitoring = True
TemperatureMonitoring = False ; Default setting off
```

Reference: OEM Package HMI

Hibernate (only for Windows XP)

The parameter HibernateMode is in the section [Miscellaneous] to set the hibernate versions.

These settings can also be made via the user interface of HMI-Advanced. To do this, select in the operator area "Start-up" → "HMI" → "System Settings".

```
[Miscellaneous]
HibernateMode = ...
```

The following values can be set:

`HibernateMode = Off`	Default: Off Hibernate save and hibernate reboot operations are not performed.
`HibernateMode = Normal`	HMI-Advance first closes all applications. After the shutdown, the control is restarted automatically.
`HibernateMode = Advanced`	Not implemented

```
; Index of task started by the regie after power on
; (may differ in some circumstances from the default task index,
; which is the task index from [TaskConfiguration] with the lowest
number.
PoweronTaskIndex = 0
```

```
[StartupConfiguration]
```

OEM software in the range from 12 to 24 can be started in this section.

The [StartupConfiguration] section is subdivided as follows:

0 to 12 is reserved for expanded functionality provided by Siemens

12 to 24 free for OEM applications

25 to 32 reserved for simulation

33 to 47 reserved for Siemens applications (in the directory ADD_ON)

Example:

Startup33 = name := ITS, Timeout :=15000

Startup34 = name := HMIVers, Cmdline := "/StartUp", Timeout :=10000

```
[TaskConfiguration]
; 3. bar
; Task16 = name := oemframe, cmdline := "sysmeter", Timeout :=
10000,
; ClassName := "SysMeter", HeaderOnTop := False, PreLoad := False
```

The 3rd horizontal softkey bar is only displayed if, from task 16 onwards, a task or the EXIT button has been assigned to it.

```
; 4th bar : located to task 96 - 103
;Task96 = name := oem, Timeout := 10000
;Task97 = name := oem, Timeout := 10000
;Task98 = name := oem, Timeout := 10000
;Task99 = name := oem, Timeout := 10000
;Task100 = name := oem, Timeout := 10000
;Task101 = name := oem, Timeout := 10000
;Task102 = name := oem, Timeout := 10000
;Task103 = name := oem, Timeout := 10000
```

The 4th horizontal softkey bar is only displayed if, from task 96 onwards, a task or the EXIT button has been assigned to it. The task 50 to task 87 entries can be used by OEMs.

```
[CommandToTask]
; only necessary for HMI invocation through NCU via command channel
plc_select = #13
Cycles = #14
```

In this section, you can specify that the HMI will be started from the NCU via the command channel.

References: Commissioning Manual "Supplement operator interface",
Chapter "Activating the dialog screen from the NC program"

A.23 SEDITOR.INI

Overview

You can modify the displays in the **multi-channel step sequence programming (option)** by making the appropriate settings or specific supplements in this file.

The sections that contain the code [*User*] allow the cycles of the machinery construction OEM and the user to be identified as "steps" and display these. In addition, other details can be specified.

To configure your own steps, settings can be made in the following sections:

[SyncMarks_User]	Expansions to incorporate synchronization marks
[USER_VARIABLES]	Variables for channels with synchronization marks
[Old_or_User_Cycles_USER]	Expansions for user cycles; any variable names can be entered here.
[ProgSupport_User]	Expansions to incorporate your own cycles

The merge mechanism is used for the settings; this means that with respect to the standard version of the ini file, only changes and supplements have to be saved in this directories add_on, oem and user.

Syntax rules

Note the following information regarding your expansions:

```
;**************************************************************
;The following rules must be used for the sections:
; - [SyncMarks_Siemens]
; - [SyncMarks_User]
; - [Old_or_User_Cycles_Siemens]
; - [Old_or_User_Cycles_USER]
;
;- You can use either *- or ?-statements.
; Do not mix these placeholders:
;
; L1* -> OK
; L1?? -> OK
; L1?1* -> NOT OK !!!
;
;- ? is replaced with numbers from 0 to 9, so the step editor
; will find the following strings:
;
; L1? -> L10, L19, L15, but NOT L1A
; LB?? -> LB23, LB99, LB00, but NOT LB1A or LBd6
;
```

```
;- You can use the ?-statement once per entry:
; L1? -> OK
; L1?? -> OK
; L23??? -> OK
; L2??3? -> NOT OK !!!
;**************************************************************
```

```
[MISC]
IconPath
```

Relative path to the directory in which the icons are stored.

The path is relative to ..\mmc2. The same directory can also be present in Add_on, oem, and user. The icons are also searched for in these directories.

Default setting: IconPath = ae\skpicto

```
EndIcon
```

Icon for the last block. This icon must be located in the directory that was specified in IconPath.

```
GCodeIcon
```

Icon for free DINCode. This icon must be located in the directory that was specified in IconPath.

```
ShowBlockNumber
```

Display of line number (e.g. N210). The number of the first line of a block is always displayed.

0: Line number is not displayed.

1: Line number is displayed.

```
TimeFormat
```

The displayed lines can be formatted in VB syntax, e.g. number of decimal places.

####0.000: Time display with 3 decimal places

```
CopySyncBuffers
```

For the synchronized view, you can specify here whether or not the empty blocks that were inserted for synchronization will be copied.

0: Do not copy empty blocks.

1: Copy empty blocks.

```
[INSERT]
TemplateForNewBlock
```

Using the key combination CTRL+I, in the edit mode, a step of a SEFORM instruction can be added. The text to be inserted can be configured here.

```
[Cache]
Enabled
```

A file can be opened faster by using the cache.

The individual data for the sequence view is stored separately and can be re-accessed, provided the part program has not been changed.

Presently, only the following setting is available:

0: Cache is not active.

1: Cache is active.

`[DefaultIcons]`

These icons are displayed if icons are not specified in the individual definitions, e.g. if an icon is not included in SEFORM(…). This icon must be located in the directory that was specified under IconPath.

```
Block
Icon for main step – SEFORM(…, 0, …)
SubBlock
Icon for sub-step – SEFORM(…, 1, …)
Sync
Icon for synchronization marks – e.g. WAITM(…)
Cycle
Icon for cycle
GeoProz
Icon for contour path
[SyncMarks_User]
```

Machine manufacturers can enter their expansions for synchronization marks here.

```
;Definition of synchronization marks (search texts)
;sync1 = WAITM*(*n,*c,*c,*c,*c,*c,*c,*c,*c,*c,*c)
; '#' : separator: text before is searched in one
; channel, text behind in all others.
; It isn't possible to synchronize this kind of
; synchronization mark!!!
; '*n': identifier for synchronization numbers
; WAITM(*n,1,2,3)
; '*c': channel WAITM(1,*c,*c,*c), if own channel
; is missing, it will be added like WAITM-strategy
;
; Add additional/language-dependent text:
; WAITM*(*n,*c,*c,*c,*c,*c,*c,*c,*c,*c,*c) "$85000 The Text"
;
;sync1_Ico = clock.bmp
; belonging icon for defined sync mark
;sync1_Chans = 1,2,4
; Project the channels to synchronize here
; e.g.: you have an M-function (M4711)
; to synchronize the channels 1, 2 and 4
; Then you can write here: syncX=1,2,4
; If you synchronize the part programs the
; channel 3 will not be considered.
; If you don't project anything here, in all
; channels must stand the M4711 call
; respective if there are *C in [SyncMarks]
; only these channels will be synchronized.
;SIEMENS
[SyncMarks_Siemens]
sync1 = WAITM*(*n,*c,*c,*c,*c,*c,*c,*c,*c,*c,*c)
sync1_Ico = clock.bmp
sync1_Chans =
;USER
[SyncMarks_User]
```

`sync1`

Definition of search text. The following syntax applies:

```
 * space holder; text in '*' must be the same in all channels.

 n identification using a number e.g. WAITM(*n,1,2,3)
 Wait marks can be assigned using this number.

 *c Channel numbers, these do not have to be in the same order,
 the user's own channel does not have to be specified
 (e.g.: WAITM(1,*c,*c,*c).

 # Channel separator character: Text in front of the separator
character is
 searched for in a channel, the text is then searched in all others.
These synchronization marks
 cannot be synchronized with one another.
```

`sync1_Ico`

Associated icon, it must be located in the directory that was specified in IconPath.

`sync1_Chans`

Here, it can be specified whether the associated synchronization marks will be searched for in certain channels only. If no entry is made, the search will be carried out in all channels or in those that are specified if *c was configured.

Example:

M4711 is a synchronization that only acts on Channels 1, 2, and 6. The entry must look like this:

```
sync2 = M4711
sync2_Ico = clock.bmp
sync2_Chans = 1,2,6
[Old_or_User_Cycles_User]
```

Machine manufacturers can enter their expansions for their own cycles here.

Examples:

```
 Cycle1 = Cycle*
 Cycle1_Type = 1
 Cycle1_Ico = cycle.bmp
 Cycle2 = Pocket*
 Cycle2_Type = 1
 Cycle2_Ico = cycle.bmp
 ...
CycleX
```

Definition of cycles without graphical support. Subprogram calls can also be entered here. A language-specific text can also be specified. X stands for a cycle number, which must be the same over three associated entries.

Syntax:	The following is displayed, e.g.:
CycleX = L100(*	"L100(2,3,4)"
CycleX = L100(* "$85001"	Language-dependent text that is stored in 85001
CycleX = L100(* "$85001/fixed Text"	Language-dependent text that is saved in 85001 + "/fixed text"

```
CycleX_Type
```

Definition of whether cycle is to be displayed as a sub-block or a main block

```
1: Sub-block
0: Main block
CycleX_Ico
```

Associated icon, it must be located in the directory that was specified in IconPath.

```
[ProgSupport_User]
```

Machine manufacturers can enter their expansions for their own cycles here.

Example:

```
;Programming Support - Siemens Standard

[ProgSupport_Siemens]
NCG_1=CYC82
NCG_2=CYC86
NCG_3=CYC83
NCG_4=CYC84
NCG_5=CYC84_1
```

```
NCG_6=CYC840_1
NCG_7=HOL1
NCG_8=HOL2
NCG_9=POS1
NCG_10=WPOS
...
;drilling

[CYC82]
;drilling
NCG_1_Label=NC1
NCG_1_Text=$80854
NCG_1_Ico=cm_poly_dr_drill.bmp
NCG_1_Level=0
NCG_2_Label=NC2
NCG_2_Text=$80855
NCG_2_Ico=cm_poly_dr_drill.bmp
NCG_2_Level=0
;reaming
NCG_3_Label=NC3
NCG_3_Text=$80853
NCG_3_Ico=cm_poly_dr_reaming.bmp
NCG_3_Level=0
NCG_4_Label=NC4
NCG_4_Text=$80852
NCG_4_Ico=cm_poly_dr_reaming.bmp
NCG_4_Level=0
…

[CYC86]
;boring
NCG_1_Label=NC1
NCG_1_Text=$80666
NCG_1_Ico=cm_poly_dr_bore.bmp
NCG_1_Level=0
NCG_2_Label=NC2
NCG_2_Text=MCALL CYCLE86
NCG_2_Ico=cm_poly_dr_bore.bmp
NCG_2_Level=0

[CYC83]
;deep hole drilling
NCG_1_Label=NC1
NCG_1_Text=$80794
NCG_1_Ico=cm_poly_dr_deepdrill.bmp
NCG_1_Level=0
```

```
NCG_2_Label=NC2
NCG_2_Text=MCALL CYCLE83
NCG_2_Ico=cm_poly_dr_deepdrill.bmp
NCG_2_Level=0
```

Explanations:

As an example, two graphically-supported cycles are shown here.

The following lines appear in place of the cycle in the NC program:

```
;NCG#CYC82#\CST.DIR\bohren.com#NC1#2#*NCG;*RO*;*HD*
;#1#1#1#1#1###"M3"####1##1#*NCG;*RO*;*HD*
CYCLE82(1,2,3,4,,0)
;#END#*NCG;*RO*;*HD*
[ProgSupport_User]
No.
```

Here, the number of the following entries must be specified.

`NCG_1`

Entries of individual search texts or name of cycles.

Generally, this is the text that is located in the NC program after ";NCG#".

In the example above, this would be "CYC82".

Because there can be several versions (e.g. NC1, NC2, etc.) for such a cycle, an additional section must be set-up in the INI file for this entry [CYC82].

```
Section for cycles (e.g.: [CYC82])
No.
```

Number of cycle versions

`NCG_1_Label`

"Name" of the version. Generally, this is the text that is found in the NC program after the path of the COM file (NC1, NC2, etc.).

`NCG_1_Text`

Text that is displayed. A fixed text, a language-specific text ($85111), or a combination of both ($85111/my additional text) can be specified.

`NCG_1_Ico`

Associated icon, it must be located in the directory that was specified in IconPath.

`NCG_1_Level`

Definition of whether cycle is to be displayed as a sub-block or a main block

```
1: Sub-block
0: Main block
```

Reference: Commissioning Manual "Supplement operator interface"

A.24 SIMTOGEO.INI

Overview

Sample file with the details required for simulation for the individual tool types.

Note

If Link is set to 0 in the [Settings] section, the other entries in SIMTOGEO.INI are redundant.

```
[SETTINGS]
; this entry is dependent on settings such as
;
; MD 18206: $MN_MM_NUM_CCS_TOA_PARAM=1
; MD 18207: $MN_MM_TYPE_CCS_TOA_PARAM[1]=4
; MD 18080: $MN_MM_TOOL_MANAGEMENT_MASK='H4'
;
; if required and / or if other oem parameters are used, the index
can be changed
; ( e.g. TC_DPCS4 if $MN_MM_NUM_CCS_TOA_PARAM=4 )
; or any other oem variable than TC_DPCS1 can be used :
;
;LINK=TC_DPC7
;
; MD 18096: $MN_MM_NUM_CC_TOA_PARAM=7
; MD 18097: $MN_MM_TYPE_CC_TOA_PARAM[7]=4
; MD 18080: $MN_MM_TOOL_MANAGEMENT_MASK='H4'
;
LINK=0 ; example LINK=TC_DPCS1
; example data sets

[500.1] ; Unique identifier (t,d) for roughing tool 1
tool_type=500 ; tool type, if not set, $TC_DP1 is
 ; used
insert_length= ; TC_DP8, plate length
holder_angle= ; TC_DP10, holder angle
reference_direction= ; TC_DP11, cutting direction
clearance_angle= ; TC_DP24, clearance angle

[500.2] ; Unique identifier (t,d) for finishing tool 1
tool_type=510 ; tool type, if not set, $TC_DP1 is ; used
insert_length= ; TC_DP8, plate length
holder_angle= ; TC_DP10, holder angle
reference_direction= ; TC_DP11, cutting direction
clearance_angle= ; TC_DP24, clearance angle
```

```
[520.1] ; Unique identifier (t,d) for plunging tool 1
tool_type=520 ; tool type, if not set, $TC_DP1 is used
insert_length= ; TC_DP8, plate length
reference_direction= ; TC_DP11, cutting direction

[530.1] ; Unique identifier (t,d) for parting tool 1
tool_type=530 ; tool type, if not set, $TC_DP1 is used
insert_length= ; TC_DP8, plate length
reference_direction= ; TC_DP11, cutting direction

[540.1] ; Unique identifier (t,d) for thread-cutting tool 1
tool_type=540 ; tool type, if not set, $TC_DP1 is used
reference_direction= ; TC_DP11, cutting direction

[160.1] ; Unique identifier (t,d) for boring-threading-milling
cutter
tool_type=160 ; tool type, if not set, $TC_DP1 is used
tip_angle= ; TC_DP10, tip angle
;TC_DPV3= ; 1=+X; -1=-X
;TC_DPV4= ; 1=+Y; -1=-Y
;TC_DPV5= ; 1=+Z; -1=-Z

[151.1] ; Unique identifier (t,d) for saw
tool_type=151 ; tool type, if not set, $TC_DP1 is ; used
width= ; TC_DP9, width
;TC_DPV3= ; 1=+X; -1=-X
;TC_DPV4= ; 1=+Y; -1=-Y
;TC_DPV5= ; 1=+Z; -1=-Z
;data set for tool_types 200-299

[200.001] ; Unique identifier (t,d)
tool_type= ; tool type, if not set, $TC_DP1 is used
tip_angle= ; TC_DP24, tip angle
;TC_DPV3= ; 1=+X; -1=-X
;TC_DPV4= ; 1=+Y; -1=-Y
;TC_DPV5= ; 1=+Z; -1=-Z
; default tool data sets reserved by SIEMENS AG for later use
; default data sets are recognized by non digit values
; examples

;[500] ; number corresponding to tool type, roughing tool
;insert_length= ; TC_DP8, plate length
;holder_angle= ; TC_DP10, holder angle
;reference_direction= ; TC_DP11, cutting direction
;clearance_angle= ; TC_DP24, clearance angle
```

```
;[510] ; number corresponding to tool type, finishing tool
;insert_length= ; TC_DP8, plate length
;holder_angle= ; TC_DP10, holder angle
;reference_direction= ; TC_DP11, cutting direction
;clearance_angle= ; TC_DP24, clearance angle
```

A.25 TASKCONF.INI

Overview

Softkeys and softkey bars for user-specific operator areas are configured in the file TASKCONF.INI.

```
[CONTROL]
MDIList=MntMMC\MntMMC.mdi
ControlFile=MntMMC\MntMMC.zus
ScreenTwips=1

[DEBUG]
;MDISize 0 = Debug, 1 = Full screen
MDISize=0
;Buttons: 0 = button invisible, 1 = button visible
Stop_Button=1
; Here you can configure a skeleton application. Set instead
; APPLICATION, the section name configured in regie.ini.
; For every section name in regie.ini with
; Task i = name:= mntmmc cmdLine:= "SektionName", ... assigned
; the section name must be configured here for an
; autonomous section.

[APPLICATION]
; HSK1.. - HSK8.. (horizontal softkeys)
; HSK9.. - HSK16.. (ETC softkeys)
; VSK1.. - VSK8.. (vertical softkeys)
; background picture (default : none)
; 1 : machine configuration
; <filename> : load bitmap from OEM directory
Picture=
; index [1,.. initial application (default : none)
StartIndex=
; 0 : permanent task (default)
; <> 0 : terminate task at recall
TerminateTask=
;=================================================;
; for each softkey (HSK1 - HSK16 / VSK1 - VSK8) ;
;=================================================;
; taskid (see regie.ini) or logical task name
; (program/<root> or ExitButton)
; !!! ExitButton, display button for shutdown of HMI-Adv !!!
HSK1Task=
; 0 : recall to calling process (default)
; <> 0 : terminate application at recall (appends
; TerminateTask to HSK1Command)
; must be supported by application !
HSK1TerminateTask=
; 0 : SwitchToChild (default)
; 1 : SwitchToTask -> switch only to task
; 2 : SwitchToTask2 -> switch to child of specified task
;(if it exists)
HSK1IsTask=
; command string evaluated by application (default : empty)
HSK1Command=
; access level for softkey (0 - 7, default : 7)
HSK1AccessLevel=
; 0 : no link to NCK
; <> 0 : link softkey to NCK
; (lock softkey while no connection to NCK)
HSK1NckLink=
; default softkey text, ignored for logical task name
HSK1SkText=
; language depending softkey text, ignored for logical task name
HSK1SkText_GR=
HSK1SkText_UK=
```

List of Abbreviations

B

B.1 Abbreviations

Abbreviation	Significance
A	Output in the PLC
ASCII	American Standard Code for Information Interchange American coding standard for the exchange of information
BAG	Mode group
OPI	Operator Panel Interface
CAD	Computer-Aided Design
CNC	Computerized Numerical Control Computerized numerical control
CR	Carriage Return
DAC	Digital-to-Analog Converter
DB	Data Block in the PLC
DBB	Data Block Byte in the PLC
DBW	Data Block Word in the PLC
DBX	Data block bit in the PLC
DDE	Dynamic Data Exchange: Dynamic Data Exchange
DIN	Deutsche Industrie Norm (German Industry Standard)
DIR	DIRectory: List
DPM	Dual-Port Memory
DOS	Disk Operating System
DRAM	Dynamic Random Access Memory
DRF	Differential Resolver Function: Differential revolver function (handwheel)
DRY	DRY run: Dry run feedrate
DW	Data word in the PLC
E	Input in the PLC
EG	Expansion unit
ESR	Extended Stop and Retract
FRAME	Data block (FRAME)
FIFO	First In - First Out: Method of storing and retrieving data in a memory.
BP	Basic program
GUD	Global User Data Global user data
HD	Hard Disk Hard disk
HMI	Human Machine Interface: Controller user interface
MSD	Main Spindle Drive
Hardware	Hardware
IBN	Commissioning
ICA	Interpolatory Compensation Interpolatory compensation

Abbreviation	Significance
INC	Increment: Increment
INI	INItializing data Initializing data
IPO	Interpolator
ISO	International Standard Organization
JOG	JOGging: Setup mode
K1 .. C4	Channel 1 to channel 4
LED	Light-Emitting Diode: Light-emitting diode
LF	Line Feed
K_v	Servo gain factor
LUD	Local User Data: Local user data
MB	Megabyte
MCP	Machine Control Panel Machine control panel (→ MCP)
MD	Machine data
MDI	Manual Data Input: Manual input
MCS	Machine coordinate system
MLFB	Machine-readable product designation
MPF	Main Program File: NC part program (main program)
MPI	Multi-Point Interface Multiport Interface
MCP	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
ZO	Zero point shift
OEM	Original Equipment Manufacturer
OP	Operator Panel: Operating equipment
PCU	Programmable Control Unit
PCMCIA	Personal Computer Memory Card International Association: Standardization for memory cards
PG	Programming device
PLC	Programmable Logic Control:
REF	REFerence point approach function
REPOS	REPOSition function
ROV	Rapid Override: Input correction
RPA	R-Parameter Active: Memory area on the 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
SKP	SKiP: Skip block
SPF	SubProgram File: Subprogram
SRAM	Static RAM (non-volatile)
GWPS	Grinding wheel surface speed

Abbreviation	Significance
SW	Software
SYF	SYstem Files System files
TEA	Testing Data Active: Identifier for machine data
TO	Tool Offset Tool offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
UFR	User frame
FD	Feed Drive
WCS	Workpiece coordinate system
TO	Tool offset
TC	Tool change
ZOA	Zero Offset Active: Identifier (file type) for work offset data

Index

SIEMENS

SINUMERIK 840Di sl/840D sl
SINUMERIK 810D/840D

Expanding the user interface

Commissioning Manual

11/2006
6FC5397-0DP10-0BA0

Safety Guidelines

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**

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

⚠ **Warning**

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the 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
Automation and Drives
Postfach 48 48
90437 NÜRNBERG
GERMANY

Order No.: 6FC5397-0DP10-0BA0
Edition 10/2006

Table of contents

1 Scope of performance

Overview

"Expand user interface" is implemented with an interpreter and configuration files that contain the descriptions of the user interfaces. "Expand user interface" 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.

The "Expand user interface" tools can be used to create user interfaces that display functional expansions designed by the machine manufacturer or end user, or simply to implement your own dialog layout. Preconfigured user interfaces supplied by Siemens or the machine manufacturer can be modified or replaced.

The interpreter is available for HMI Embedded sl, ShopMill and ShopTurn on NCU as well as HMI Advanced.

Parts programs, for example, can be edited on user interfaces created by users. Dialogs can be created directly on the control system.

Prerequisites

An additional graphics program is needed to produce graphics/display images. For HMI Embedded sl you need the application disk and the Paint Shop Pro tool (http://www.jasc.com). The tool box supplied contains configuration examples for new dialogs. You can also use these examples as a template for creating your own dialogs.

Use

You can implement the following functions:

1. Display dialogs containing the following elements:
   - Softkeys
   - Variables, tables
   - Texts and Help texts
   - Graphics and Help displays
2. Open dialogs by:
   - Pressing the (start) softkeys
   - Selection on the PLC
3. 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
4. Initiate operations in response to the following actions:
   - Displaying dialogs
   - Input values (variables)
   - Select 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:
   - Subroutines
   - File functions
   - PI services
   - External functions (HMI Advanced)
8. Apply protection levels according to user classes

Supplementary Conditions

The following conditions must be met:

- It is only possible to switch between dialogs within one HMI operating area.
- In the case of HMI Advanced, user, setting and machine data are initialized on request.
- User variables may not have the same names as system or PLC variables.
- The dialogs activated by the PLC form a separate operating area for HMI Advanced (similar to measuring cycle displays).

Note

The programming support functions described in the chapter of the same name and the user interfaces for Siemens cycles have been created with the system tools for Expand user interface. As a result, they can be modified as required by the machine manufacturer or end user within the scope described in this chapter.

See also:

You can find details of the configuration files in the chapter "Configuring environment".

2 Programming

2.1 Getting started

2.1.1 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 be set up by the user.

An ASCII editor (e.g., Notepad or the HMI editor) is used to create configuration files.

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:

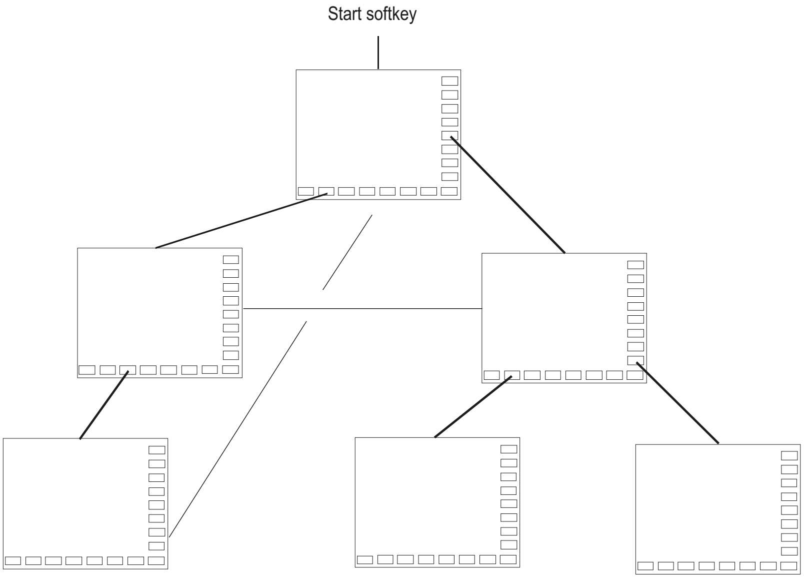


Figure 2-1 Menu tree

Start softkeys

One or more softkeys (start softkeys), which are used to initiate your own operating sequences, are defined in one of the specified configuration files.

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 user interfaces and return to the standard application.

You can use the <RECALL> key to close new user interfaces if you have not configured this key for any other task.

Calling your own dialogs from PLC

Dialogs can be selected via the PLC as well as via softkeys. An interface (in DB19) is available for signal exchange between the PLC and HMI.

See also

Structure of the interface (Page 4-12)

Configuring start softkeys (Page 6-5)

2.1.2 Configuration and start files

Overview

Every application uses permanent (HMI Embedded sl, ShopMill, and ShopTurn on NCU) or preset (HMI Advanced) start softkeys, which can be be used to access newly generated dialogs. Additional start softkeys can be configured with HMI Advanced.

Other files:

In the event of "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 operation mode in a separate configuration file.

Creating configuration file as ASCII file

Dialogs can contain, for example, the following elements:

- Input/output fields (variables) with
  - Short text
  - Graphic text
  - Text for units
- Images
- Tables
- Softkey menus

Search sequence for configuration files

- HMI Embedded sl on NCU

  For HMI Embedded sl as well as ShopMill and ShopTurn on NCU, the system searches for the configuration files by accessing the relevant directories of the CF card.

  A setting can be made in file COMMON.COM in the standard cycle directory/user cycle directory of HMI Embedded sl to specify whether a configuration file search should be carried out each time access is attempted (only relevant when the dialogs are being set up directly on the control) or if the file already located and buffered should be reused (corresponds to the standard operating scenario).
- HMI Advanced

  With HMI Advanced, the system starts by searching for the configuration files in the user cycle directory and then looks in the manufacturer cycle directory followed by the standard cycle directory.

See also

Search function principle (Page 6-10)

2.1.3 Structure of configuration file

Overview

A configuration file consists of the following elements:

- Description of the start softkeys
- Definition of dialogs
- Definition of variables
- Description of the blocks
- Definition of a softkey menu

Example

```
//S (START)                             ; Definition of the start softkey (optional)
....
//END
//M (.....)                             ; Definition of the dialog
DEF .....                               ; Definition of variables
LOAD                                    ; Description of the blocks
...
END_LOAD
UNLOAD
...
END_UNLOAD
ACTIVATE
...
END_ACTIVATE
...
//END
//S (...)                               ; Definition of a softkey menu
//END
```

2.1.4 Troubleshooting (log book)

Overview

The log book is a file (Error.com) to which error messages generated by syntax interpretation are written. The operator himself must set up the file in the comment directory (HMI Advanced).

Example

```
DEF VAR1 = (R)
DEF VAR2 = (R)
LOAD
VAR1 = VAR2 + 1                ;  Error message in log book, as VAR2 has no value.
```

Syntax

The system does not start to interpret 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 form")
PRESS(HS6)
  LM("Screen form1")
END_PRESS
//END

//M(Screen form1)□
  DEF Var1=(R)
//END
```

Content of ERROR.COM

If "Expand user interface" detects errors when interpreting the configuration files, these errors will be written to the ERROR.COM ASCII file.

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

If the dialog was created using the PC test environment, then the error file will be stored in the folder referenced by environment variable RAMDISK (HMI Embedded sl).

The ERROR.COM file is created only when errors actually occur during interpretation of the configuration files.

Storage path of the ERROR.COM file:

- With HMI Embedded sl: In the /tmp/hmiemb folder on the CF card
- With HMI Advanced: In the \DH\COM.DIR\ folder

This file is deleted each time HMI Embedded sl / HMI Advanced is restarted.

Displaying the ERROR.COM file

HMI Advanced:

- Call the editor in the "Services" or "Startup" operating area.

HMI Embedded sl:

- "Setup" operating area → "HMI" → "Editor" → "Temp drive" (4th softkey on vertical softkey bar; although this softkey is not displayed with the manufacturer password, it will respond). (It is also displayed with the system password.)
- Select ERROR.COM.
- Press the <INPUT> key.
- With the “File functions” softkey, the file can also be copied to a PC by means of a Windows network drive.

2.2 Structure and elements of a dialog

2.2.1 Defining a dialog

Definition

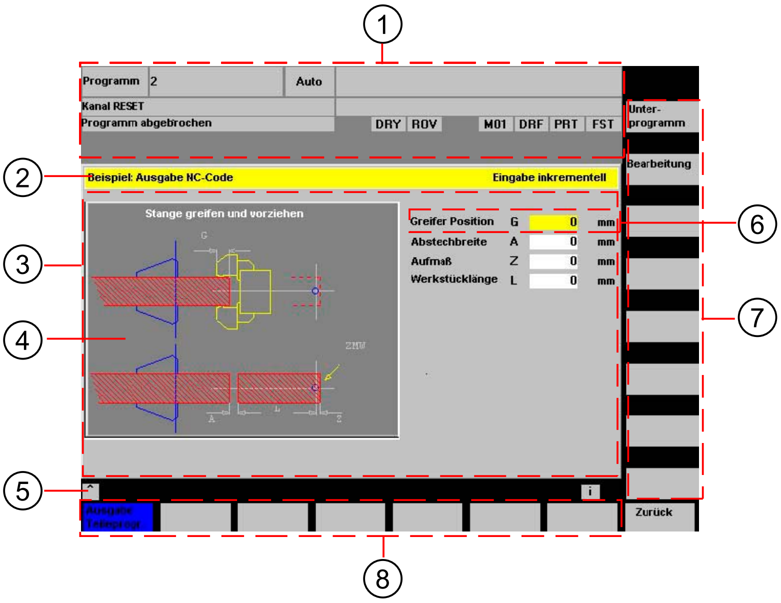
A dialog is part of a user interface consisting of a display line, dialog elements and/or graphics, an output line for messages and 8 horizontal and 8 vertical softkeys.

Dialog elements are:

- Variables
  - Limits
  - Default setting of variables
- Help display
- Texts
- Attributes
- System or user variable
- Position of short text
- Position of input/output field
- Colors
- Help (HMI Advanced only)

Dialog properties:

- Header
- Graphic
- Dimension
- System or user variable
- Graphic position
- Attributes



Greifer Position G 0 mm
Abstechbreite A 0 mm
Aufmaß Z 0 mm
Werkstücklänge L 0 mm

Figure 2-2 Structure of the dialog

1 Machine status display ("header")
2 Header line of the dialog with header and long text
3 Dialog
4 Graphic
5 Displaying messages
6 Dialog element
7 8 vertical softkeys
8 8 horizontal softkeys

Overview

The definition of a dialog (definition block) is basically structured as follows:

Definition block	Comment	Chapter reference
`//M...`	;Dialog start identifier	

`DEF Var1=...`
`...`
	;Variables	See chapter "Variables"

`HS1=(...)`
`...`
	;Softkeys	See chapter "Softkey menus"

`PRESS(HS1)`
`  LM...`
`END_PRESS`
	;Method start identifier
	  ;Actions
	;Method end identifier
		See chapter "Methods"

`//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.

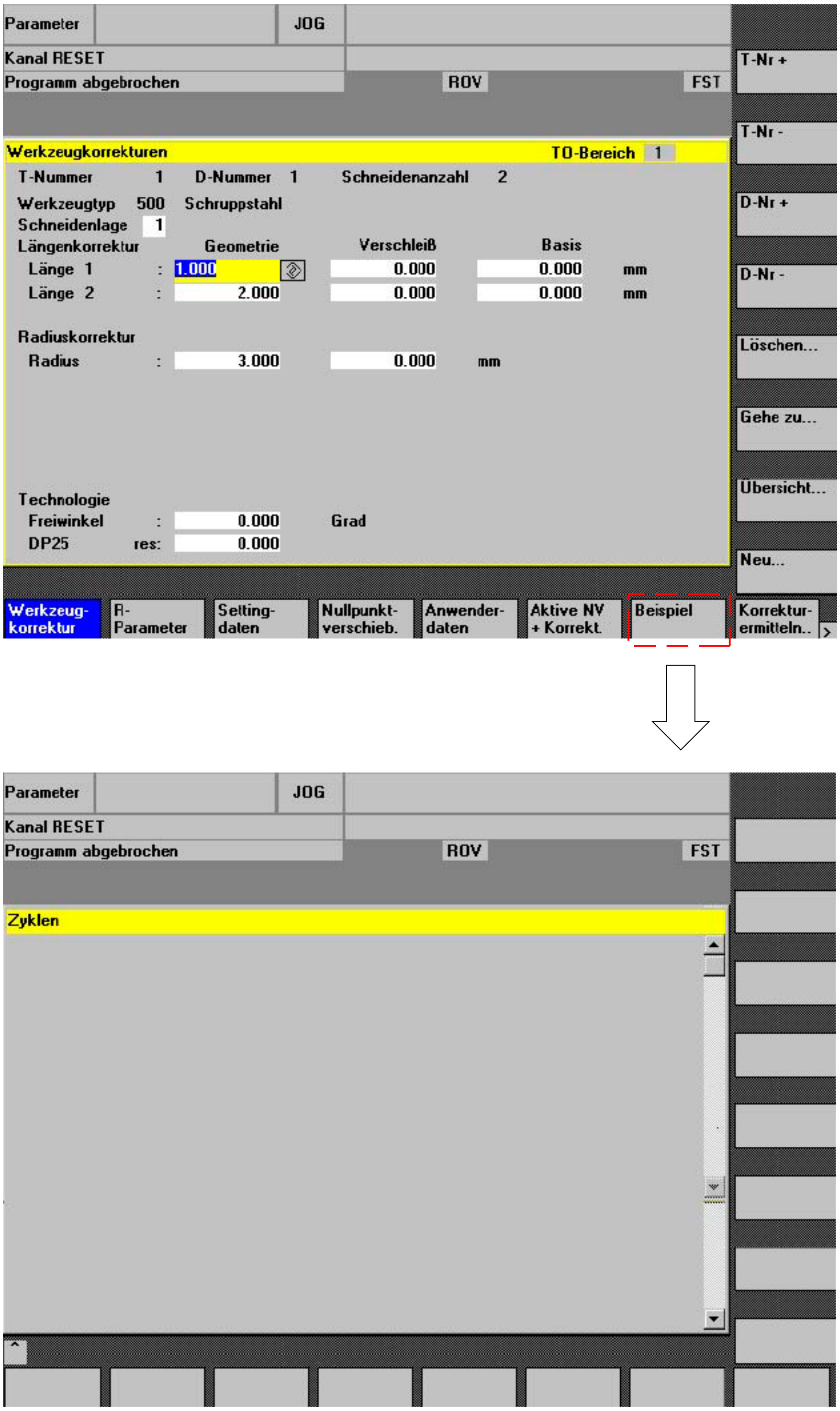
2.2.2 Example Opening the Dialog

Programming

A new dialog is called via start softkey "Example" from the "Parameters" operating area.

```
//S(Start)
HS7=("Example", ac7, se1)

PRESS(HS7)
  LM("Screen form1")
END_PRESS

//END
//M(Screen form1/"Cycles")
HS1=("")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")
VS2=("")
VS3=("")
VS4=("")
VS5=("")
VS6=("")
VS7=("")
VS8=("")
...                                         ;  Methods
//END
```

Result



Parameter JOG
Kanal RESET
Programm abgebrochen ROV FST

Werkzeugkorrekturen TO-Bereich 1

T-Nummer 1 D-Nummer 1 Schneidenanzahl 2
Werkzeugtyp 500 Schruppstahl
Schneidenlage 1

Längenkorrektur		Geometrie	Verschleiß	Basis	
Länge 1	:	1.000	0.000	0.000	mm
Länge 2	:	2.000	0.000	0.000	mm

Radiuskorrektur				
Radius	:	3.000	0.000	mm

Technologie			
Freiwinkel	:	0.000	Grad
DP25	res:	0.000	

Parameter JOG
Kanal RESET
Programm abgebrochen ROV FST

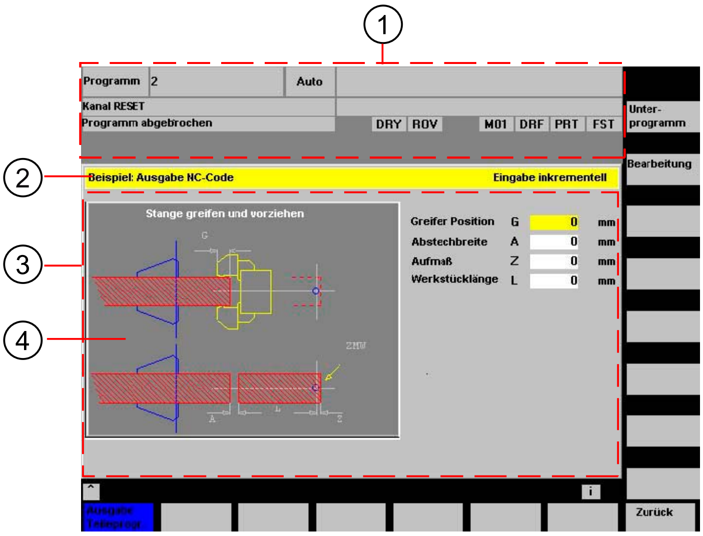
Zyklen

Figure 2-3 Example Call "Cycles" dialog with the start softkey "Example"

2.2.3 Dialog properties

Description

The properties of the dialog are defined in the start identifier line of the dialog.



Programm 2 Auto
Kanal RESET
Programm abgebrochen

Beispiel: Ausgabe NC-Code   Eingabe inkrementell

Stange greifen und vorziehen

Greifer Position G 0 mm
Abstechbreite A 0 mm
Aufmaß Z 0 mm
Werkstücklänge L 0 mm

Figure 2-4 Dialog properties

1 Machine status display ("header")

2 Header line of the dialog with header and long text

3 Dialog

4 Graphic

Programming

Syntax: //M(Identifier/[Header]/[Graphic]/[Dimension]/[System or user variable]/[Graphic position]/[Attributes])

Description: Defines a dialog

Parameters:

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 right-hand side, 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 cursor position can be assigned to the NC or PLC via the system or user variable. The first variable has the 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 right-hand side), in relation to the upper left-hand corner of the dialog. The minimum distance from the top is 18 pixels. The entries are separated by a comma.

Attributes — The specifications of the attributes are separated by a comma.

Possible attributes are:

CMx — Column mode: Column alignment

CM0 Default: 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.

CB0 Default: 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.

System — The "System" property can be read during runtime:

0: HMI_Embedded

1: HMI_Advanced

Accessing the dialog properties

Read and write access is provided to the following dialog properties within Methods (e.g., PRESS block)

- Hd = Header
- Hlp = Help display
- Var = System or user variable

Example

```
//S(Start)
HS7=("Example", se1, ac7)

PRESS(HS7)
  LM("Screen form1")
END_PRESS

//END
//M(Screen form1/"Example 2 : Display
graphics"/"MCP.BMP")
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
```

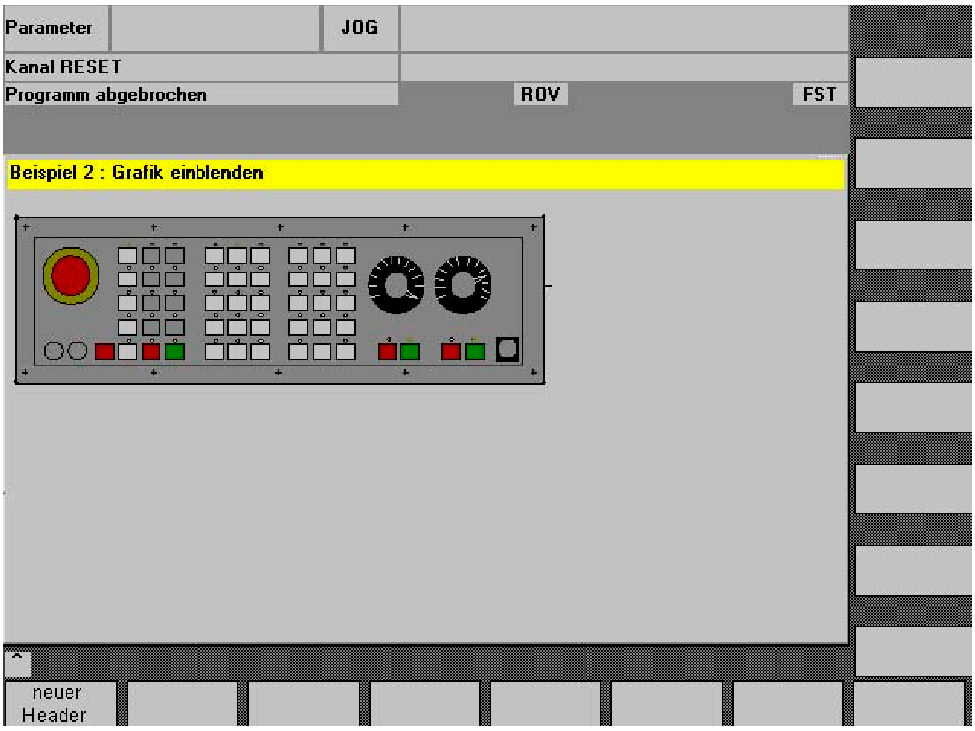


Figure 2-5 "Example 2:Display graphics"

See also

Using display images/graphics (Page 2-17)

Language-dependent text (Page 6-6)

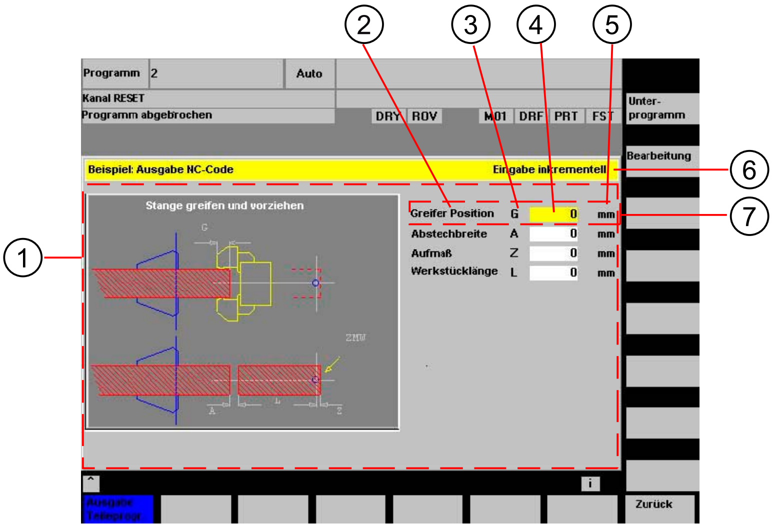
2.2.4 Dialog elements

Dialog element

The term "dialog element" refers to the visible part of a variable, i.e., short text, graphics text, input/output field and unit text. 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 only in the active dialog. Properties are assigned to a variable when it is defined. The values of dialog properties can be accessed within Methods (e.g., a PRESS block).



Beispiel: Ausgabe NC-Code   Eingabe inkrementell

Greifer Position G 0 mm
Abstechbreite A 0 mm
Aufmaß Z 0 mm
Werkstücklänge L 0 mm

Figure 2-6 Elements of a dialog

1	Main body of the dialog
2	Short text
3	Graphic text
4	Input/output field
5	Text for units
6	Header line of the dialog with header and long text
7	Dialog element

Programming - Overview

The single parameters to be separated by commas are enclosed in round parentheses:

`DEF` *Identifier* =	**Identifier = Name of variable**		
	Variable type	→	1
	/[Limits or toggle field or table identifier]	→	2
	/[Default]	→	3
	/[Texts (Long text, Short text|Image, Graphic text, Units text)]	→	4
	/[Attributes]	→	5
	/[Help display]	→	6
	/[System or user variable]	→	7
	/[Position of short text]	→	8
	/[Position of input/output field(Left, Top, Width, Height)]	→	9
	/[Colors]	→	10
	/[Help] (HMI Advanced only)	→	11

See also

Dialogs with multiple columns (Page 2-15)

Variable properties (Page 2-19)

2.2.5 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)
...
```

When creating dialogs with multiple columns, the options of the hardware being used should be taken into consideration, e.g., HMI Embedded sl supports up to 10 columns and 60 DEF instructions.

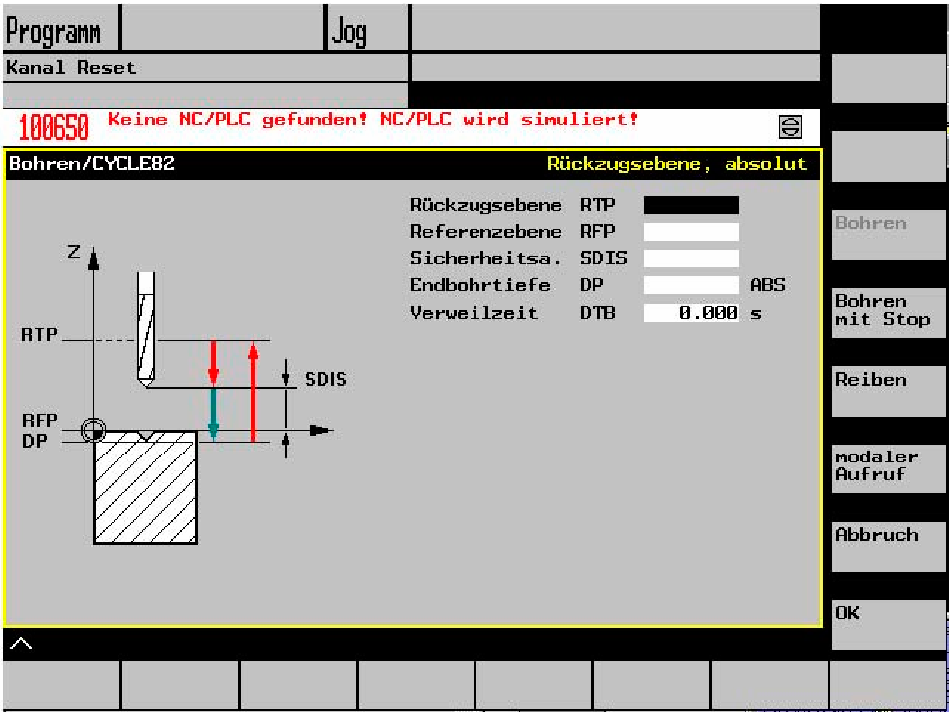
2.2.6 Unifying the dialog appearance

Boundary condition

If HMI Advanced is installed together with ShopMill or ShopTurn on one operator panel, then the two systems have different font types.

HMI Advanced has "proportional fonts", while JobShop products and HMI Embedded sl have a "fixed font".

If "Expand user interface" is used on HMI Embedded sl and HMI Advanced, dialogs that are defined identically will appear differently by default.



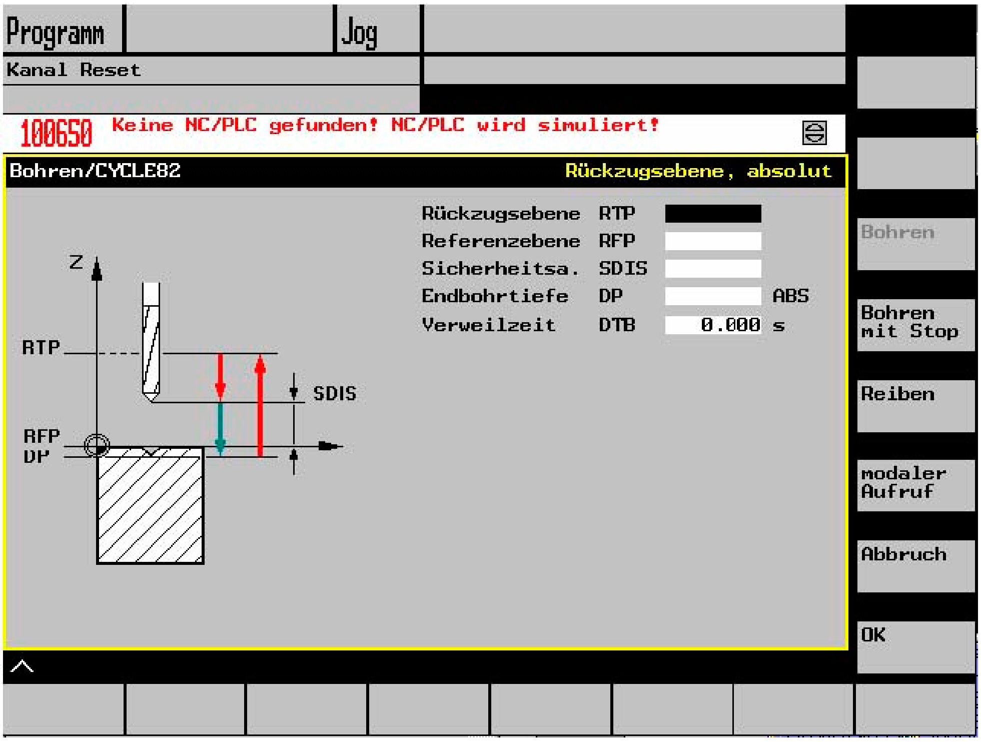
100650 Keine NC/PLC gefunden! NC/PLC wird simuliert!

Bohren/CYCLE82 Rückzugsebene, absolut

```
Rückzugsebene  RTP
Referenzebene  RFP
Sicherheitsa.  SDIS
Endbohrtiefe   DP           ABS
Verweilzeit    DTB   0.000  s
```

Figure 2-7 Appearance in HMI Embedded sl

Below you can see how, despite being configured identically, the same screen appears differently under HMI Advanced.



100650 Keine NC/PLC gefunden! NC/PLC wird simuliert!

Bohren/CYCLE82 Rückzugsebene, absolut

```
Rückzugsebene  RTP
Referenzebene  RFP
Sicherheitsa.  SDIS
Endbohrtiefe   DP           ABS
Verweilzeit    DTB   0.000  s
```

Figure 2-8 Appearance in HMI Advanced

Setting for the same screen appearance

In order for the appearance in HMI Advanced to tally with that in HMI Embedded sl, the font is changed to a fixed font in the configuration file WIZARD.INI:

```
[FONT]
FixedFont=1
```

The text will be displayed in the current language in fixed font format. The only exception is text for the softkey labels. Unlike the HMI Embedded sl input field, the HMI Advanced input field contains an input mode pictogram. This means it is smaller than the HMI Embedded sl field.

As you can see above, the combination of short text, graphic text, input field, and unit text is approximately the same length and contains the same information in both screens.

However, individual fields can be positioned differently in the two HMI versions depending on the length of the text to be displayed and the text field within which it will be contained.

To enable a common configuration file to be produced for both systems, the SYSTEM dialog property has been created. The runtime environment can be scanned in the SYSTEM dialog property. This enables a common configuration to be used for both target systems.

2.2.7 Using display images/graphics

Use of graphics

There are two display categories:

- 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.

Maximum size	System
560 * 326 pixels	HMI Advanced/HMI Embedded sl
688 * 376 pixels	PCU 50 with OP 012

Creating screen images for HMI Embedded sl

Create the screen images using, e. g., the MS Paint application.

HMI Embedded sl can handle converted graphics as well as BMP files. Graphics created using the conversion program BMP2BIN can continue to be used. For BMP files, the colors are converted online. For this purpose, the files containing the color table (syscol.col, sysbw.col, etc.) have been extended to include the [BMP] section. This section allocates a color of a BMP pixel to a color in HMI Embedded.

Storage locations

HMI Embedded sl automatically finds the resolution of the connected monitor and searches for the desired display image first in the associated resolution directory. If it is not found there, HMI Embedded sl searches for the display image in the next smaller resolution directory until – if it doesn’t find the display image earlier – it reaches directory ico640.

Procedure

Procedure when integrating the bitmaps (customer displays):

1. Integration on HMI Embedded sl
2. Generate BMP files using Microsoft Paint, Version 4.0 or higher
3. Use the arj.exe supplied (Version 2.41) to archive the BMP files into archive CUS.ARJ or, alternatively, archive each BMP file in a separate archive with the file name of the BMP file and extension ".BM_"

   For example:

   – Several files per archive:

     arj a cus.arj my_file1.bmp my_file2.bmp my_file3.bmp my_file4.bmp

   – One file per archive:

     arj a my_file1.bm_ my_file1.bmp

Limits

Archiving is optional. However, it must be noted that the CF card has space for at least 10 MB.

A bitmap can be assigned to any softkey if the softkey text is preceded by 2 backslashes. The text behind the backslashes is then interpreted as the name of the file that contains the bitmap.

For example, in ALUC.TXT 85000 0 0 "\\mybitmap.bmp"

Note

The display colors might not appear in the same format in HMI Embedded sl and in Paint.

Display building, HMI Advanced

Graphic programs can be freely selected provided that they are capable of generating the prescribed formats. Graphics, display images and Help displays can have the following format:

- Bitmap (BMP)
- Windows Metafile (WMF)
- Icon (ICO)

"Display in display"

You can insert further displays in the background (Help display), even as a function of the values of variables.Like for softkeys you can assign a graphics file to a display field.

Examples:

- Short text field is overlaid with graphics:

  DEF VAR1=(S///,"\\image1.bmp" ////160,40,50,50)   ;Type S irrelevant
- Toggle field overlaid with graphics, the graphic being selected with a PLC flag byte:

  DEF VAR1=(IDB/*1="\\image1.bmp",2="\\image2.bmp"//,$85000/wr1//"MB[0]"//160,40,50,50)

The height of the displayed graphic (of the I/O field) is specified as the fourth parameter in respect of the positions. You can also define bitmaps in fields (short text, I/O field).

See also

Search function for images (Page 6-14)

List of colors (Page A-5)

2.3 Variables

2.3.1 Variable properties

Variable value

The most basic 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 Identifier = Variable value	
Description:	Variable value val (value)	
Parameters:	Identifier:	Name of the variable
	Variable value:	Value of variable

Example	`VAR3 = VAR4 + SIN(VAR5)`
	`VAR3.VAL = VAR4 + SIN(VAR5)`	

Variable status

The "Variable status" property can be scanned during runtime to find out whether a variable has a valid value. This property can be read and written with the value FALSE = 0.

Programming

Syntax:	Identifier.**vld**	
Description:	Variable status vld (validation)	
Parameters:	Identifier:	Name of the variable
		The result of the scan can be:
	FALSE =	invalid value
	TRUE =	valid value

Example

```
IF VAR1.VLD == FALSE
  VAR1 = 84
ENDIF
```

Variable: Changing properties

When a change is made, a new value is assigned to the variables in the notation *Identifier.Property = Value* . The expression to the right of the equality sign is evaluated and assigned to the variable or variable property.

Examples:

Identifier.**ac** = Access level	(ac: access level)
Identifier.**al** = Text alignment	(al: alignment)
Identifier.**bc** = Background color	(bc: background color)
Identifier.**fc** = Foreground color	(fc: foreground color)
Identifier.**fs** = Font size	(fs: font size)
Identifier.**gt** = Graphic text	(gt: graphic text)
Identifier.**hlp** = Help display	(hlp: help)
Identifier.**htx** = Help text	(htx: help text)
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.**st** = short text	(st: short text)
Identifier .**typ** = Variable type	(typ: type)
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)

2.3.2 Programming variables

Programming

The parameters of the variables are explained briefly in the following overview. A comprehensive description can be found in subsequent chapters.

Parameter	Description
1 Variable type	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
2 Limits	Limit value MIN, limit value MAX Default: Empty 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".
Toggle field	List with predefined entries in the I/O field: The list starts with a * and the entries are separated by commas. The entries can be assigned a value. 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. Default: none
Table identifiers	Identifiers of a table containing NCK/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.
3 Default setting	If a default setting has not 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, i.e., the status of the variable is "not calculated". Default: No default
4 Texts	The sequence is specified. Instead of a short text, an image can also be displayed. Default: 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

Parameter	Description	
5 Attributes	The attributes influence the following properties: • Input mode • Access level • Alignment of short text • Font size • Limits • Response when dialog is opened in terms of CHANGE block The attributes are separated by commas and appear in any order. The attributes are not valid for toggle fields. A definition can be made for each component.	
	Input mode	wr0: Input/output field invisible, short text visible wr1: Read (no focus possible for input) wr2: Read and write (line appears in white) wr3: wr1 with focus wr4: All variable elements invisible, no focus possible 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). Default: wr2
	Access level	empty: Can always be written ac0...ac7: Protection levels 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 al1: Right-justified al2: centered Default: al0
	Font size	fs1: Default font size (8 pt.) fs2: Double font size Default: fs1 The clearances between the lines is defined. With the default font size, 16 lines can 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. Default: Determined by specified limits li0: No check li1: Check with respect to min. li2: Check with respect to max. 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.

Parameter	Description
	cb0: The CHANGE block defined for this variable is processed when the dialog is opened (default setting). Multiple attributes are separated by commas. cb1: The CHANGE block defined for this variable is then only processed if the value of the variable changes.
6 Help display	Help display file: Name of the PDF-file Default: 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.
7 System or user variable	An item of system or user data on the NC/PLC can be assigned to the variable. The system or user variable appears in double quotation marks. References: Parameter Manual, "List of System Variables" /PGA1/
8 Position of short text	Position of short text (distance from left, distance from top, width) 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.
9 Position of input/output field	Position of input/output field (distance from left, distance from top, width, height) 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.
10 Colors	Foreground color, background color: The colors are separated by a comma. Color settings are only relevant to the input/output field; colors cannot be specified for the other texts. Range of values: 1...10 Default: Foreground color: Black, background color: white The default colors of the input/output field are determined by the Write mode:
	wr0: Foreground and background color: Window background color wr1: Text color: Black, background color for window wr2: Text color: Black, background color: white wr3: As wr0 wr4: As wr1 wr5: As wr2
11 Help (HMI Advanced only)	Help file: Path to the PDF-file Index: Index in the Help text file to the Help text Help text: Help text for display in the Help text file

Parameter	Description
	The data is separated by commas, the sequence is defined. The Help file and Help text appear in double quotation marks. The PDF files must be filed in folder CUS.DIR\hlp.dir or CST.DIR\hlp.dir. PDF and text files corresponding to one another must have the same name. The name of the PDF file must be written in capital letters in the txt file. Several help references can be configured in a loop (Help loop) for each dialog element, i.e., the references are called up in succession and when the last reference closes, the first is displayed again. If a second or subsequent Help call is linked to the same file/index/Help text, the entries do not have to be made. Help is displayed when the cursor is positioned on this field and the information button is pressed.

2.3.3 Detailed information about the variable type [1]

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
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. It can be scanned using the ISNUM or ISSTR functions. The VARIANT type is mainly suited to write either variable names or numerical values to the NC code.

Programming

The data type of variables can be checked:

Syntax: **ISNUM** *(VAR)*

Parameters: 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)*

Parameters: 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 ERROR.COM file.

  Example

  ```
  Var1.typ = "IBW"
  Var2.typ = "R3"
  ```

Number representation

Numbers can be represented in either binary, decimal, hexadecimal or exponential notation. Numerical values in binary, hexadecimal and exponential notation must be enclosed in single quotation marks:

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.

2.3.4 Detailed information about the toggle field [2]

Description

The toggle field extension function can be used to display 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.

Programming

Syntax: **DEF Identifier =(Variable type /+ $Text number | * value="\\display"[,value="\\image2.bmp"][, ...]**

/[Default]

/[Texts(Long text, Short text, Graphic text, Units text)]

/[Attributes]

/[Help display]

/[System or user variable]

/[Position of short text]

/[Position input/output field(Left, Top, Width, Height)]

/[Colors]

/[Help])

Description: When the dialog is opened, the content of text number $85015 is displayed in the input/output field. Default value 15 is entered in system variable DB90.DBB5. If the value saved in system variable DB90.DBB5 changes, then the displayed text number $(85000 + <DB90.DBB5>) is recalculated in response to every change.

Parameters:

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.

Example

```
DEF VAR1=(IB/+ $85000/15////"DB90.DBB5")
```

Variable toggle field

It is possible to assign a variable toggle field to a dialog element, i.e., when the toggle key is pressed, a value configured in a CHANGE method is assigned to the variable.

An asterisk * is entered in the Limits or Toggle Field property to identify a variable toggle field when a variable is defined.

Example: `DEF VAR1=(S/*)`

Toggle-field-dependent displays

The toggle field is overlaid with alternating graphics: If the value of the flag byte is 1, "image1.bmp" will appear. If it is 2, "image2.bmp" will appear.

```
DEF VAR1=(IDB/*1="\\image1.bmp",
                2="\\image2.bmp"//,$85000/wr1//"MB[0]"//160,40,50,50)
```

The size and position of the image is defined under "Position of input/output field (left, top, width, height)".

2.3.5 Detailed information about the default setting [3]

Overview

A variable can assume various states depending on whether a default value, or a system or user variable, or both, has been assigned to the variable field (I/O field or toggle field). (Not calculated: Toggling is not possible until a valid value is assigned to the variable).

Scope of the default settings

If...			Then...
Field type	Default setting	System or user variable	Reaction of field type
I/O field	yes	yes	Write default value to system or user variable
	No	yes	Use system or user variable as default value
	Error	yes	Not calculated, system or user variable is not written into/used.
	yes	No	Default setting
	No	No	Not calculated
	Error	No	Not calculated
	yes	Error	Not calculated
	No	Error	Not calculated
	Error	Error	Not calculated
Toggle	yes	yes	Write default value to system or user variable
	No	yes	Use system or user variable as default value
	Error	yes	Not calculated, system or user variable not written/used
	yes	No	Default setting
	No	No	Default = first toggle field element
	Error	No	Not calculated
	yes	Error	Not calculated
	No	Error	Not calculated
	Error	Error	Not calculated

2.3.6 Detailed information about the position of the short text [8] and position of the input/output field [9]

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 settings 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.

If no positions have been specified for any dialog elements, the default setting is applied. By default, the column width for the short text and input/output field is calculated for each line based on 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.

2.3.7 Detailed information about on Help [11] (HMI-Advanced only)

Description

The Help loop for a dialog element can be extended or deleted during runtime. The Help loop can be extended as required by calling the function repeatedly.

Programming

Syntax:	**ADDHTX** (Identifier, Help file, Index, Help text)	
Description:	Extends the Help loop	
Parameters:	Identifier	Name of the variable whose Help loop is to be expanded.
	Help file:	Path specification of the file (PDF format)
	Index:	Index in the Help text file to the Help text
	Help text:	Help text for display in the Help text file
Example	`ADDHTX(VAR1,"C:\OEM\HLP\MYHLP.PDF",15,"Machine data")`	

Syntax: **CLRHTX** (Identifier)

Description: Deletes the Help loop

Parameters:
Identifier — Name of the variable whose Help loop is to be deleted
Help file: — Path specification of the file (PDF format)
Index: — Index in the Help text file to the Help text
Help text: — Help text for display in the Help text file

Example

```
CLRHTX(VAR1)
```

2.3.8 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

Parameters: Identifier: — Name of Help variables

Example

```
DEF OTTO  ;Definition of a Help variable
```

Syntax:
Identifier.val = *Help variable value*
Identifier = *Help variable value*

Description: A value is assigned to a Help variable in a method.

Parameters:
Identifier: — Name of Help variables
Help variable value: — Content of the Help variables

Example

```
LOAD
  OTTO = "Test"              ;  Assign the value "Test" to the Otto Help variable
END_LOAD
LOAD
  OTTO = REG[9].VAL          ;  Assign value of register to the Help variable
END_LOAD
```

Calculation with variables

Variables are calculated every time you exit an input/output 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.,

```
Var1 = Var5 + SIN(Var2)
Otto = PI * Var4
```

Addressing system variables indirectly

A system variable can also be addressed indirectly, i.e., as a function of another variable:

```
PRESS(HS1)
  AXIS=AXIS+1
  WEG.VAR="$AA_DTBW["<<AXIS<<"]"   ;Address axis address via
variable
END_PRESS
```

Changing the softkey label

Example

```
HS3.st = "New Text"   ;Change softkey label
```

2.3.9 Example 1: Assigning the Variable Type, Texts, Help, Colors properties

Example 1

Assigning the Variable Type, Texts, Help, Colors properties

DEF Var1 = (R///,"Actual value",,"mm"//"Var1.bmp"////8,2)

Variable Type:	REAL
Limits or toggle field entries:	none
Default:	none
Texts:	
Long text:	None
Short text:	Actual value
Graphic text:	none
Unit text:	mm
Attributes:	none
Help display:	Var1.bmp
System or user variable:	none
Position of short text:	No data, i.e., default position
Position of input/output field:	No data, i.e., default position

DEF Var1 = (R///,"Actual value",,"mm"//"Var1.bmp"////8,2)		
Colors:		
	Foreground color:	8
	Background color:	2
Help:		none

2.3.10 Example 2: Assigning the Variable Type, Limits, Attributes, Short Text Position properties

Example 2

Assigning the Variable Type, Limits, Attributes, Short Text Position properties

DEF Var2 = (I/0,10///wr1,al1///,,300)		
Variable Type:		INTEGER
Limits or toggle field entries:		MIN: 0
		MAX: 10
Default:		none
Texts:		none
Attributes:		
	Input mode	read-only
	Alignment of short text	Right-justified
Help display:		none
System or user variable:		none
Position of short text:		
	Distance from left	None
	Distance from top	None, i.e., default distance from top left
	Width:	300
Position of input/output field:		No data, i.e., default position
Colors:		No data, i.e., default
Help:		none

2.3.11 Example 3: Assigning the Variable Type, Default, System or User Variable, Input/Output Field Position properties

Example 3

Assigning the Variable Type, Default, System or User Variable, Input/Output Field Position properties

DEF Var3 =(R//10////"$R[1]"//300,10,200//"Help.pdf",1,"Help1")	
Variable Type:	String
Limits or toggle field entries:	none
Default:	10
Texts:	none
Attributes:	none
Help display:	none
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
Colors:	No data, i.e., default
Help:	In file Help.pdf, Help with the Help text "Help1" is called on the page with the index 1 when the `<i>` button is pressed.

2.3.12 Examples relating to toggle field, Help call and image display

Example 4

Various entries in the toggle field:

Limits or toggle field entries:

```
DEF Var1 = (I/* 0,1,2,3)
DEF Var2 = (S/* "In", "Out")
DEF Var3 = (B/* 1="In", 0="Out")  ;1 and 0 are values, "In" and "Out" are displayed.
DEF Var4 = (R/* ARR1)             ;ARR1 is the name of an array.
```

Example 5 (HMI Advanced only)

Several Help calls per dialog element:

```
DEF Var5 = (R//////////"Help1.pdf",1,"Help1",,2,"Help2","Help3.pdf",3,)
```

Variable Type:	REAL	
Limits or toggle field entries:	none	
Default:	none	
Texts:	none	
Attributes:	none	
Help display:	none	
System or user variable:	none	
Position of short text:	None	
Position of input/output field:	none	
Colors:	No data, i.e., default	
Help:	1. Entry in the Help loop	
	Help file:	HELP1.PDF
	Index:	1
	Help text:	Help1
	2. Entry in the Help loop	
	Help file:	HELP2.PDF
	Index:	2
	Help text:	Help2
	3. Entry in the Help loop	
	Help file:	HELP3.PDF
	Index:	3
	Help text:	Help3

Example 6

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.bmp" ////160,40,50,50)
```

Variable Type:	VARIANT
Limits or toggle field entries:	none
Default:	none
Texts:	None
Attributes:	none
Help display:	none
System or user variable:	none

DEF VAR6= (V///,"\\image1.bmp" ////160,40,50,50)		
Position of short text:		image1.bmp
Position of input/output field		
	Distance from left:	160
	Distance from the top:	40
	Width:	50
	Height:	50
Colors:		No data, i.e., default
Help:		none

2.3.13 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.

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) << "th error"
            ;Result: "This is the 11th error"
```

- Treatment of numerical values:

```
VAR13.VAL = "Error " << VAR14.VAL << ": " << $T80001 << $T80002
         ;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]
```

2.3.14 CURPOS variable

Description

Using the CURPOS variable, it is possible to display or manipulate the position of the cursor in the active input field of the current dialog. The variable indicates how many characters are located 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 block. If the value of CURPOS changes, then a jump is made to the CHANGE block and the instructions contained there are executed.

2.3.15 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
                                     automatically assigned the version of the
                                     recompiled code.
```

2.3.16 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 = Programming support (the dialog was called by programming support)

2 = Programming support + default setting from the previous dialog (sub-dialog)

3 = Programming support + recompilation

4 = Programming support + recompilation with generated comments, with # sign

5 = Programming support + recompilation with generated comments, without # sign

Example

```
IF ENTRY == 0
  DLGL("The dialog was not called during programming")
ELSE
  DLGL("The dialog was called during programming")
ENDIF
```

2.3.17 ERR variable

Description

Variable ERR can be used to check whether the preceding lines have 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                      ;   Scan to check whether value has been found
                                        in array
  VAR5 = "Error accessing array"
                                    ;   If the value has not been found in the
                                        array, the value "Error accessing array" is
                                        assigned to the variables.
ELSE
  VAR5 = "All OK"                   ;   ;If the value has been found in the array,
                                        the value "All OK" is assigned to the
                                        variables.
ENDIF
```

2.3.18 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                    ;  Scan to ascertain whether error has occurred
   IF FILE_ERR == 1                ;  Scan 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 = "Wrong 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
```

2.3.19 FOC variable

Description

The FOC variable can be used to control the input focus (the current active input/output field) in a dialog. Responses to cursor left, right, up and down movements, as well as PGUP, PGDN, are predefined and cannot be modified.

Note

The FOC function may not be initiated as a result of a navigation event. The cursor position may only be changed in softkey PRESS blocks, CHANGE blocks, 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 will be assigned to Variable 1.
FOC = 3                           ;  The input focus will be assigned to the 3rd
                                     dialog element with WR ≥ 2.
```

2.3.20 S_CHAN variable

Description

The S_CHAN variable can be used to determine the number of the current channel for display or evaluation purposes.

2.4 Complex dialog elements

2.4.1 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)*
	(a/b...)
	(c/d...)
	...
	//END	

Description:	Defines array	
Parameters:	Identifier	Name of array
	a, b, c, d	Values of array 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
```

2.4.2 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. The ERR variable is also set to TRUE if a search concept cannot be found.

Programming

Syntax: Identifier [Z,[M[,C]]].val or
Identifier [Z,[M[,C]]]

Description: Access to one-dimensional array with only one column

Syntax: Identifier [S,[M[,C]]].val] or
Identifier [S,[M[,C]]] or

Description: Access to one-dimensional array with only one line

Syntax: Identifier [Z,S,[M[,C]]].val or
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: 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: Search in line 0	Value from line Z and column found
3	Search concept: Search in column 0	Search concept: Search in line 0	Value from line and column found
4	Search concept: Search in column S	Column index of search column	Line index
5	Line index of search line.	Search concept: 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 larger 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 found in the search line or search column. If the search concept cannot be found, the value 0 or an empty string is output and the error variable ERR is set to TRUE.

2.4.3 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. 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.

2.4.4 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
	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
```

2.4.5 Table grid (grid)

Definition

In contrast to the array, the values of a table grid (grid) are continually updated. This involves a tabular representation of the values of system variables that can be addressed using one block in one channel.

Assignment

A variables definition is assigned to the table-elements definition via a table identifier:

- The variables definition determines the values to be displayed and the table-elements definition determines the on-screen appearance and assignment. The table grid takes the properties of the input/output fields from the variables definition line.
- The visible area of the grid is determined by the width and height of the I/O field. Any lines or columns than cannot be seen can be displayed by scrolling horizontally and vertically.

Description

The variables definition will contain a reference to a table description:

`DEF` *Identifier* =	**Identifier = Name of variable**		
	Variable type	→	1
	/[Limits or toggle field or table identifier]	→	2
	/[Default]	→	3
	/[Texts (Long text, Short text|Image, Graphics text, Units text)]	→	4
	/[Attributes]	→	5
	/[Help display]	→	6
	/[System or user variable]	→	7
	/[Position of short text]	→	8
	/[Position input/output field(Left, Top, Width, Height)]	→	9
	/[Colors]	→	10
	/[Help] (HMI Advanced only)	→	11

Table identifier [2]

Identifiers of a table containing NCK/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.

System or user variable [7]

This parameter remains empty for table grids, because the column definition lines contain detailed information about the variables to be displayed. The table description can be provided in a dynamic format.

2.4.6 Defining table grids

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 table grids		
Parameters:	Table identifiers	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
	No. 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.	
	Fixed line attribute	1:	Active
		0:	Not active
	Fixed column attribute	1:	Active
		0:	Not active

2.4.7 Defining columns

Description

For table grids, it is advisable to use variables with an index. For PLC or NC variables, the index number is of significance with one or more indices.

The values displayed in a grid can be modified directly by the end user within the restrictions of the rights granted by the attributes and within any limits defined.

Programming

Syntax:	*(Type/Limits/Empty/Long text,column header/Attributes/Help display/ System or user variable/Column width/Offset1, Offset2, Offset3)*	
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	The PLC or NC variables should be entered in double quotation marks.
	Column width	Entry in pixels.
	Offset	The increments to be used to accumulate each index in order to fill the column are specified in the assigned offset parameter: • Offset1: Step width for the 1st index • Offset2: Step width for the 2nd index • Offset3: Step width for the 3rd index

Variable of type STRING

If the variable is a STRING type variable, the length must be specified in the type, e.g., `DEF CHAN STRING [16] TEXT[41].`

The column definition for the CHAN variable, therefore, starts, e.g., (S16/...).

Column header from text file

The column header can be entered as text or text numbers ($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 display (hlp) and
- OPI-Variable (var).

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"`

Column properties cannot be read in the LOAD block.

The wr, ac and li attributes can be specified for column definitions.

See also

Dialog elements (Page 2-13)

List of accessible system variables (Page A-7)

2.4.8 Focus control in the table grid

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 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
```

2.4.9 Example Defining columns

Overview

The next three examples show the assignment between cells and PLC variables in a standard table (table type=0).

Example 1:

The first line shows the column headers:

```
//G(MB_TAB/0/4/,1)
(I///,"MB 1 to MB 3"///"MB1"/100/1)
(I///,"MB 4 to MB 6"///"MB4"/100/1)
```

Result:

MB 1 to MB 3	MB 4 to MB 6
Value(MB1)	Value(MB4)
Value(MB2)	Value(MB5)
Value(MB3)	Value(MB6)

Example 2:

Offset > 1 in the column definition; this results in the following lines and column assignments:

```
//G(MB_TAB/0/4/,1)
(I///," MB1, MB3, MB5"///"MB1"/100/2)
(I///," MB2, MB4, MB6"/// "MB2"/100/2)
```

Result: In each line, the index of the variable is increased by the offset (=2).

MB1, MB3, MB5	MB2, MB4, MB6
Value(MB1)	Value(MB2)
Value(MB3)	Value(MB4)
Value(MB5)	Value(MB6)

Example 3:

Offset and index number in the column definition:

- In the first column, the first variable index for each line is increased by 1: Offset 1 = 1
- In the first column, the second variable index for each line is increased by 1: Offset 2 = 1

```
//G(MB_TAB/0/4/,1)
(IB///,"M1.1, M2.1, M3.1"/// "M1.1"/100/1)
(IB///,"M1.1, M1.2, M1.3"/// "M1.1"/100/,1)
```

Result:

M1.1, M2.1, M3.1	M1.1, M1.2, M1.3
Value(M1.1)	Value(M1.1)
Value(M2.1)	Value(M1.2)
Value(M3.1)	Value(M1.3)

Other options:

- Ascending numbers can be entered in the first column:

  Example: `(I///,"Line"///"0"/60/1)`
- Consecutive texts from the language files can be entered in the first column:

  Example: `(S///,"Line"///"$80000"/60/1)`

2.4.10 Example Loading different table grids

Description

In this example the "dummygrid" table is first assigned to variable "VAR1". Depending on the content of the R-parameter R[0], either the "grid1" or "grid2" table will be loaded in the LOAD block. The tables are defined in the same file as the variable "VAR1".

```
//M(SCREEN FORM1/"GRID")
DEF VAR1=(R/% dummygrid//////200,75,300,85)
HS1=("")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")
VS2=("")
VS3=("")
VS4=("")
VS5=("")
VS6=("")
```

```
VS7=("EXIT",ac7,se1)
VS8=("")

LOAD
  REG[0] = RNP ("$R[0]")
    IF (REG[0] == 0)
       LG ("grid1", "var1")
   ELSE
       LG ("grid2", "var1")
   ENDIF
END_LOAD

PRESS(VS7)
   EXIT
END_PRESS

//END

//G(grid1/0/5/1,1)                              ;  (Name/Type/Lines...)
(R///"Long text1","R1 to R4"/wr2//"$R[1]"/80/1)  ; 1. Column, header "R1 to R4", from R1 with Offset1
(R///"Long text2","R5 to R8"/wr2//"$R[5]"/80/1)  ; 2. Column from R5
(R///"Long text3","R9 to R15"/wr2//"$R[9]"/80/2) ; 3. Column from R9 with Offset2, R9 11 13 15
//END

//G(grid2/0/5/1,1)                              ;  (Name/Type/Lines...)
(R///"Long text1","R1 to R4"/wr2//"$R[1]"/60/1)  ; 1. Column, header "R1 to R4", from R1 with Offset1
(R///"Long text2","R5 to R8"/wr2//"$R[5]"/60/1)  ; 2. Column from R5
(R///"Long text3","R9 to R15"/wr2//"$R[9]"/60/2) ; 3. Column from R9 with Offset2, R9 11 13 15
(R///"Long text4","R9 to R15"/wr2//"$R[9]"/60/2) ; 3. Column from R9 with Offset2, R9 11 13 15
//END

//G(dummygrid/0/5/1,1)                          ;  (Name/Type/Lines...)
(R///"Long text1","R1 to R4"/wr2//"$R[1]"/80/1)  ; 1. Column, header "R1 to R4", from R1 with Offset1
(R///"Long text2","R5 to R8"/wr2//"$R[5]"/80/1)  ; 2. Column from R5
//END
```

2.5 Softkey menus

2.5.1 Softkey description

Overview

The names of the soft keys 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 term softkey menu is used to refer to all the horizontal and vertical softkeys displayed on a screen form. In addition to the existing softkey menus, it is possible to define other menus, which partially or completely overwrite the existing menus.

The definition of a softkey menu (softkey menu definition block) is basically structured as follows:

Definition block	Comment	Chapter reference
`//S...`	;Start identifier of softkey menu	
`HSx=...`	;Define softkeys	

`PRESS(HSx)`
`  LM...`
`END_PRESS`
	;Method start identifier
	;Actions
	;Method end identifier
		See chapter "Methods"

`//END`	;End identifier of softkey menu	

2.5.2 Define softkey menu

Description

Properties are assigned to softkeys during definition of the softkey menu.

Programming

Syntax:	//S*(Identifier)*	;Start identifier of softkey menu
	...	
	//END	;End identifier of softkey menu
Description:	Defines softkey menu	
Parameters:	Identifier	Name of softkey menu

Syntax:	SK = *(Text[, Access level][, Status])*	
Description:	Define softkey	
Parameters:	SK	softkey, e.g., HS1 to HS8, VS1 to VS8
	Text	Enter text
	Display file name	"\\my_pic.bmp" or via separate text file $85199, e.g., with the following text in the (language-specific) text file: 85100 0 0 "\\c:\pic\my_pic.bmp". Size of image, which can be displayed on a softkey: max. 80 x 34 pixels
	Access level	ac0 to ac7 (ac7: default)
	Status	se1: visible (default) se2: disabled (gray text) se3: displayed (last softkey used)

Note

Enter %n in the softkey text to create a line break.

A maximum of 2 lines of 10 characters each are available in HMI Advanced and 2 lines of 9 characters each in HMI Embedded sl.

Assigning the security level

The operator can only access information on the level for which he is authorized and all lower levels.

The meanings of the different protection levels are as follows: ac0 is the highest protection level and ac7 the lowest.

Security level	Locked by	range
ac0	Password	Siemens
ac1	Password	Machine manufacturer
ac2	Password	Service
ac3	Password	User
ac4	Keylock switch position 3	Programmer, machine setter
ac5	Keylock switch position 2	Qualified operator
ac6	Keylock switch position 1	Trained operator
ac7	Keylock switch position 0	Semi-skilled operator

Example

```
//S(Menu1)                      ;  Start identifier of softkey menu
HS1=("NEW",ac6,se2)             ;  Define softkey HS1, assign the label "OK", security
                                   level 6 and the status "disabled"
HS3=("\\bild1.bmp")             ;  Assign a graphic to the softkey
HS5=("Exit")
VS2=("Subscreen")
VS3=($85011, ac7, se2)
VS7=("Abort", ac1, se3)         ;  Define softkey HS1, assign the label "Cancel",
                                   security level 1 and the status "displayed".
VS8=("OK", ac6, se1)               Define softkey VS8, assign the label "OK", security
                                   level 6 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("Screen form21")           ;  Load dialog
END_PRESS                       ;  Method end identifier
//END                           ;  Softkey menu end identifier
```

<RECALL> key

⋀

In addition to the horizontal and vertical softkeys, a <RECALL> key is also available.

In contrast to the softkeys, the RECALL key does not need to be defined. The "Status" and "Access Level" properties can be assigned to the key during runtime.

If you do not assign any action (function, calculation of variables, property changes) to <RECALL>, you can use it to exit newly configured user interfaces and return to the standard application.

Example

```
PRESS(RECALL)
  RECALL.ac = 1
  LM("Screen form5")
END_PRESS
```

2.5.3 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""		;Softkey with label
	SK.ac = Access level		;Softkey with security level
	SK.se = Status		; Softkey with status
Description:	Assigns properties		
Parameters:	Text		Label text in inverted commas
	Access level		Range of values: 0...7
	Status	1:	visible and operator-controllable
		2:	disabled (gray text)
		3:	displayed (last softkey used)

Example

```
//S(Start)
HS7=("Example", ac7, se1)

PRESS(HS7)
  LM("Screen form3")
END_PRESS

//END

//M(Screen form3/"Example 3 : Graphics and
softkeys"/"MST.BMP")
HS1=("")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")
VS2=("")
```

```
VS3=("")
VS4=("")
VS5=("")
VS6=("")
VS7=("")
VS8=("OK", AC7, SE1)
PRESS(VS8)
  EXIT
END_PRESS
//END
```

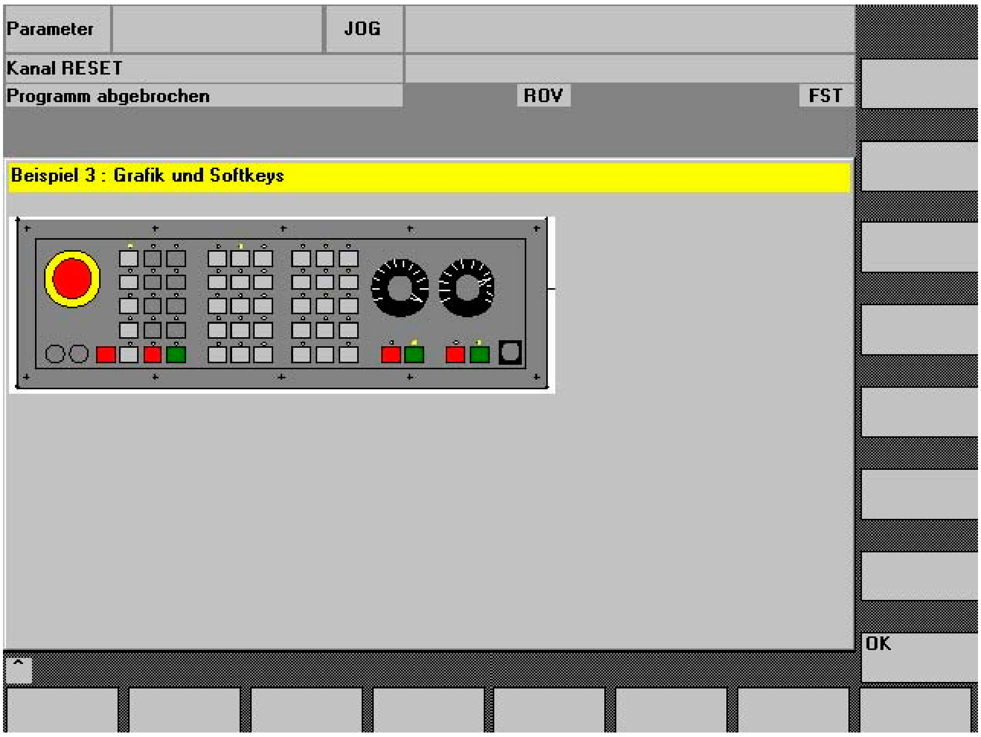


Figure 2-9 Example 3: Graphics and softkeys

2.5.4 Defining the start softkey

Dialog-independent softkey

Start softkeys are dialog-independent softkeys that, rather than being called from a dialog, have been configured **before** the first new dialog. In order to access the start screen 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
```

2.5.5 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.
- The **"EXE" function** can be used on HMI Advanced to call a program created as an application with the HMI Advanced OEM package or to start the Free Contour Programming application. With HMI Embedded sl, "EXE" can only be used to start the Free Contour Programming application.

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 form
                                       2 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 menu
END_PRESS
PRESS (HS2)                                      ; empty PRESS method
END_PRESS
```

See also

Load Softkey (LS) (Page 2-88)

Load Mask (LM) (Page 2-87)

2.6 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	Chapter reference
`PRESS(HS1)`	;Method start identifier	

`LM...`
`  LS...`
	;Functions	See chapter "Functions"

`Var1.st = ...`	;Changing properties	see chapter "Softkey menu" and chapter "Dialog elements"

`Var2 = Var3 + Var4`
`...`
`EXIT`
	;Calculation with variables	See chapter "Defining variables"

`END_PRESS`	;Method end identifier	

2.6.1 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)* ... END_CHANGE	
Description:	Changes the value of a specific variable	
Parameters:	Identifier	Name of the variable

Example

```
DEF VAR1=(S//////"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()
...
END_CHANGE

Description: Changes any variable value

Parameters: - None -

Example

```
CHANGE()
  EXIT                        ; If any of the variable values change, the dialog will
                                be terminated.
END_CHANGE
```

2.6.2 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 may only be moved in softkey PRESS blocks, CHANGE blocks, etc. Responses to cursor movements are are predefined and cannot be modified.

Note

Within the FOCUS block, it is not possible to select a different variable, nor can a new dialog be loaded.

Programming

Syntax:	FOCUS ... END_FOCUS
Description:	Positions the cursor
Parameters:	- None -

Example

```
FOCUS
  DLGL("The focus has been placed on variable" << FOC << ".)       °  °
END_FOCUS
```

2.6.3 LOAD GRID

Description

The table description can be made available dynamically within the LOAD block 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.

2.6.4 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 ... 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
```

2.6.5 UNLOAD

Description

The UNLOAD method is executed before a dialog is unloaded.

Programming

Syntax:	UNLOAD ... END_UNLOAD
Description:	Unload
Parameters:	- None -

Example

```
UNLOAD
  REG[1] = VAR1                     ;  Save variable in register
END_UNLOAD
```

2.6.6 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)*
...
END_OUTPUT

Description: Outputs variables in the NC program.

Parameters: Identifier Name of OUTPUT method

Block numbers and skip identifiers

The OUTPUT block must not contain line numbers or skip identifiers if you wish to keep the line numbers and hide markings 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
```

2.6.7 PRESS

Description

The PRESS method is executed when the corresponding softkey is pressed.

Programming

Syntax: PRESS*(softkey)*
...
END_PRESS

Identifiers: Pressing a softkey

Parameters:	Softkey	Name of softkey: HS1 - HS8 and VS1 - VS8	
	RECALL	<RECALL> key	
	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
```

2.6.8 Example Version management with OUTPUT blocks

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")   ;  Expansion with effect from Version 1
```

When writing the OUTPUT block, you can specify which variables are written, with reference to a particular version identifier.

Example

```
OUTPUT(NC1)                    ;  Only the variables of the original version are
                                  made available in the OUTPUT block.
OUTPUT(NC1,1)                  ;  The variables of the original version and the
                                  expansions with version identifier 1 are made
                                  available in the OUTPUT block
```

The OUTPUT block for the original version 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 block 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
```

2.7 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.

Subroutines

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.

PI services

The PI_SERVICE function can be used to start PI Services (Program Invocation Services) from the PLC in the NC area.

External functions (only HMI Advanced)

Additional, user-specific functions can be integrated by means of external functions. External functions are stored in a DLL file and identified by an entry in the definition lines of the configuration file.

See also

List of PI services (Page A-17)

External functions (only HMI Advanced) (Page 2-105)

2.7.1 Activate Program (AP)

Description

The AP (Activate Program) function transfers a file from the passive HMI file system to the active NC file system. The file is loaded into the NC and enabled and then deleted in the HMI file system. With HMI Embedded sl, this function has the same effect as Set enable.

Programming

Syntax:	AP("*File*")	
Description:	Transfers a file from the passive HMI file system to the active NC file system	
Parameters:	File	Complete path name of HMI file to be transferred

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 code from the OUTPUT method to file
                                       \MPF.DIR\MESSEN.MPF
END_PRESS
PRESS(HS8)
  D_NAME = "\MPF.DIR\MESSEN.MPF"
  AP(D_NAME)                        ;  Load file into NC
END_PRESS
```

2.7.2 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*)
	SUB*(Identifier)*
	END_SUB
	[SUB(*Identifier*)
	...
	END_SUB]
	...
	//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
```

2.7.3 Subprogram call (CALL)

Description

The CALL function can be used to call 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:	Subroutine call	
Parameters:	Identifier	Name of subprogram

Example

```
//M(SCREEN FORM1)
VAR1 = ...
VAR2 = ...
LOAD
  ...
  LB("PROG1")                              ;   Load block
  ...
END_LOAD
CHANGE()
  ...
  CALL("UP1")                              ;   Call subroutine and execute
  ...
END_CHANGE
...
//END
```

2.7.4 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.

Up to 29 variables, each separated by a comma, can be checked. A character length of 500 must not be exceeded.

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, softkey 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 error-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 invalid, "VAR1
                                  and VAR2 are not OK" appears in the dialog line
ENDIF
```

2.7.5 Copy Program (CP)

Description

The CP (Copy Program) function copies files within the HMI file system or NC file system.

Note

With HMI Embedded on NCU, copying is only possible within the NC file system.

Programming

Syntax:	**CP***("Source file", "Target file")*	
Description:	Copies a file	
Parameters:	Source file	Complete path to the source file
	Target file	Complete path data of the target file

Example

```
CP("\MPF.DIR\CFI.MPF ", "\spf.dir\cfi.nc")
```

2.7.6 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:

- HMI Embedded sl: approx. 50
- HMI Advanced: approx. 100

Programming

Syntax:	**DLGL***("String*")	
Description:	Outputs text in the dialog line	
Parameters:	String	Text, which is displayed in the dialog line

Example

```
IF Var1 > Var2
  DLGL("Value too large!")       ; The text "Value too large!" appears in the dialog
                                   line if variable1 > variable2.
ENDIF
```

2.7.7 Delete Program (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	
Parameters:	File	Complete path name of file to be deleted

Example

```
DP("\MPF.DIR\CFI.MPF")
```

2.7.8 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
```

2.7.9 Execute (EXE)

Description

The EXE function can be used on HMI Advanced to call a program created as an application with the HMI Advanced OEM package or to start the Free Contour Programming application.

With HMI Embedded sl, EXE can only be used to start the Free Contour Programming application.

Note

The EXE function is only available within the parts program editor. In order to start the program, its task index must be entered in the application file under [CHILDS] as it appears in the REGIE.INI file.

Programming

Syntax:	EXE*(Program name)* ; HMI Advanced	
	EXE*(GPROC)* ; HMI Embedded sl	
Description:	Executes program	
Parameters:	Program name	Name of the program that is to be executed

Example

```
PRESS(VS3)
  EXE(GPROC)                ;    Start GPROC.EXE (free contour programming)
END_PRESS
```

2.7.10 Exist Program (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	
Parameters:	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 result of the scan can be: • M = File is stored on HMI • N = file is stored on NC • Blank string = The file neither exists on the HMI nor on the NC

Example

```
EP("\MPF.DIR\CFI.MPF", VAR1)            ; Check whether file CFI.MPF exists in the
                                          HMI file system.
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
directory")
  ENDIF
ENDIF
```

2.7.11 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 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.
                                   Transfer variables POSX, POSY and DIAMETER in
                                   doing this.
  DLGL("Screen form2 ended")     ; On returning from screen form2, the following
                                   text appears in the dialog line of screen form
                                   1: 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_DIAMETER variable to the DIAMETER
                                   variable. The variable POSY retains its current
                                   value.
END_PRESS
...
//END
```

2.7.12 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
```

2.7.13 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: Generates an NC code

Parameters:

Identifier — Name of OUTPUT block from which code is generated

Target file — Path name of target file for HMI or NC file system

If the target file is not specified (only possible within programming support system), 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.

1: Do not create comments in the generated code.

Note: This code cannot be recompiled.

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 code from OUTPUT method to file
                                               \MPF.DIR\MESSEN.MPF:
                                               Cycle123(123, -6)
                                               M30
END_PRESS
```

Recompile

- No entry for target file:

  The GC function can only be used in the Programming Support system 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 specification of a target file under "Expand user interface", an error message is output when it is executed.

- Entry for target file:

  The code generated from the OUTPUT block is transferred to 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 (HMI Advanced only). User comment lines (information required to recompile code) are not set up, i.e., the code cannot be recompiled.

Special considerations for target file specification

In principle, there are three different ways of specifying a target file:

- NC notation: /_N_MPF_DIR/_N_MY_FILE_MPF

  Only possible with HMI Embedded sl.

  The file is created in the MPF directory on the NC.

- DH notation: /MPF.DIR\MY_FILE.MPF

  Possible with both HMI Advanced and HMI Embedded sl

In the case of HMI Embedded sl, the target file specification is converted into NC notation and the file is created on the NCU.

In the case of HMI Advanced or HMI Embedded WIN32, the file is created in the data management path.

- DOS notation: d:\abc\my_file.txt or \\RemoteRechner\files\my_file.txt

  Possible with both HMI Advanced and HMI Embedded sl

  The file is written to the specified directory on the hard disk or on the specified PC, provided that the directory is available on the hard disk or on a remote PC.

  In the case of HMI Embedded sl this notation can only be used to write to the RAM drive or to a networked PC provided that a network connection has been configured.

Note

Invalid variables generate a blank string in generated NC code and an error message in the log book when they are read.

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

See also

Recompile (Page 2-96)

2.7.14 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 array from 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.

2.7.15 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 Default setting = Current configuration file

Example

```
LOAD
  LB("PROG1")                     ;  Block "PROG1" is searched for in the current
                                     configuration file and then loaded.
  LB("PROG2","XY.COM")            ;  Block "PROG2" is searched for in the
                                     configuration file XY.COM and then loaded.
END_LOAD
```

2.7.16 Load Mask (LM)

description.

The LM function can be used to load a new dialog.

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 dialog

Parameters:

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 disappears; 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.

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. EXIT will end the sub-dialog and return to the point at which the master dialog was interrupted.

In the master dialog, the UNLOAD block is not processed during the interruption.

VARx — Requirement: MS1

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

```
PRESS(HS1)
  LM("SCREEN FORM2","CFI.COM",1, POSX, POSY, DIAMETER)
                              ;  Interrupt screen form1 and open screen form2:
                                 Variables POSX, POSY and DIAMETER are transferred
                                 in doing this.
  DLGL("Screen form2 ended")  ;  On returning from screen form2, the following text
                                 appears in the dialog line of screen form 1:
                                 Screen form2 ended.
END_PRESS
```

2.7.17 Load Softkey (LS)

Description

The LS function can be used to display another softkey menu.

Programming

Syntax:	LS*("Identifier"[, "File"][, Merge])*	
Description:	Displays softkey menu	
Parameters:	Identifier	Name of softkey menu
	File	Path (HMI file system or NC file system) to the configuration file Default: Current configuration file
	Merge	
	0:	All existing softkeys are deleted; the newly configured softkeys are entered.
	1:	(Default setting) Only the newly configured softkeys overwrite the existing softkeys. The other softkeys (= standard softkeys from HMI or ShopMill/ShopTurn) remain, and retain their functionality and text.

Example

```
PRESS(HS4)
  LS("Menu2",,0)                ; Menu2 overwrites the existing menu. All existing
                                  softkeys are deleted
END_PRESS
```

Notice

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 block and will not react if navigation keys are pressed (PU, PD, SL, SR, SU, SD).

See also

Functions for start softkeys (Page 2-60)

2.7.18 Passivate Program (PP)

Description

The PP (Passivate Program) function transfers a file from the active file system on the NC to the passive file system on HMI Advanced. Once the PP function has been executed, the file is no longer present in the active file system of the NC. With HMI Embedded sl, this function has the same effect as Delete enable.

Programming

Syntax:	PP*("File*")	
Description:	Transfers a file from the active NC file system to the passive HMI Advanced file system.	
Parameters:	File	Complete path name of NC file to be transferred

Example

```
PP("\MPF.DIR\MESSEN.MPF")
```

2.7.19 Read NC/PLC (RNP), Write NC/PLC (WNP)

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. 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. If the value is a String type, it must be written in double quotation marks.

Example

```
WNP("DB20.DBB1",1)                        ;     Write PLC variable
```

2.7.20 Multiple Read NC PLC (MRNP)

Description

This MRNP command can be used to transfer several system or OPI variables in a single register access. This access method is significantly faster than reading via individual access attempts. The system or OPI variables must be included on an MRNP command of the same area.

The areas of the system or OPI 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
Parameters:	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. The following therefore applies: The value of the first variable is located in REG[Register index]. The value of the second variable is located in REG[Register index + 1], etc.

Notice

It should be noted that the number of registers is restricted and the list of variables cannot exceed 500 lines.

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].
```

Reading display machine data:

Display machine data can be read with RNP ($MM...) within the LOAD block.

General read/write access to display machine data is not possible using the "Expand user interface" function.

Note

User variables may not have the same names as system or PLC variables.

NC variable

All machine data, setting data and R parameters are available, but only certain system variables (see list in Appendix). In HMI Advanced, you can find the accessible system variables under operating area "Parameters"/"System variables"/"Edit view"/"Insert variable".

All global and channel-specific user variables (GUDs) can be accessed. but 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

PLC data	
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
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

2.7.21 REFRESH

Description

The REFRESH function can be called in all blocks. It has no parameters.

Method of operation:

- All active variable content (input/output fields) in the display range are output again with the background and foreground.
- All active and visible short description texts, graphic texts and unit texts are output again without clearing the text background first.

Programming

Syntax:	**REFRESH**
Description:	Updates content of input/output fields and output of text
Parameters:	- None -

2.7.22 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 a blank string.

Note

Registers may not be used directly in OUTPUT blocks for generating NC code.

Programming

Syntax:	REG*[x]*	
Description:	Defines register	
Parameters:	x	Register index with x = 0...19; Type: REAL or STRING = VARIANT 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* or *Identifier* = *Register value*	
Description:		
Parameters:	Identifier	Name of 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:	Status is a read-only property.	
Parameters:	Identifier	Name of 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.
```

2.7.23 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 assigned to the variable, and the subprogram
                                     ends at this point.
  ENDIF
  VAR1.val="Otto"                 ;  If the variable value ≠ Otto, the value "Otto"
                                     is assigned to the variable.
END_SUB                           ;  End of subroutine
//END                             ;  Block end
```

2.7.24 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 parts program
                           (X101 → X222)
```

The following value is displayed for VAR1 in the input dialog: `VAR1 = 222`

See also

Generate code (GC) (Page 2-82)

2.7.25 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")
```

2.7.26 Select Program (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")*	
Identifiers:	Select Program	
Parameters:	"File"	Complete path name of NC 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 code from the OUTPUT method to
                                        file \MPF.DIR\MESSEN.MPF
END_PRESS
PRESS(HS8)
 AP("\MPF.DIR\MESSEN.MPF")           ;  Load file into NC
 SP("\MPF.DIR\MESSEN.MPF")           ;  Select file
END_PRESS
```

2.7.27 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

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 blank.
	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 from string1 into string2. Enter 0 to start searching at the beginning of string2.
	String1	Character that is being searched for.

String2 Chain of characters in which the search is being made

Direction (optional) Direction in which the search is being made

0: From left to right (default setting)

1: From right to left

0 is returned if string1 does not occur in string2.

Example

```
DEF VAR01
DEF VAR02

LOAD
  VAR01="HELLO/WORLD"
  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 string to be processed

length Number of characters that are to be read out

Example

```
DEF VAR01
DEF VAR02
LOAD
 VAR01="HELLO/WORLD"
  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 string to be processed
length	Number of characters that are to be read out

Example

```
DEF VAR01
DEF VAR02
LOAD
 VAR01="HELLO/WORLD"
  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 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="HELLO/WORLD"
  VAR02=LEFT(VAR01,4,4)                   ;  Result = "LO/W"
END_LOAD
```

REPLACE Function: Replacing characters

Syntax:	**REPLACE***(string, FindString, ReplaceString [, start [, count]])*	
Description:	The REPLACE function 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 the FindString.
Return Value:		

string = Blank string	Copy of string
FindString = Blank string	Copy of string
ReplaceString = Blank string	Copy of string, in which all occurrences of FindString are deleted
start > Len(String)	Blank string
count = 0	Copy of string

2.7.28 PI services

Description

The PI_SERVICE function can be used to start PI Services (Program Invocation Services) from the PLC in the NC area.

General programming

Syntax:	**PI_SERVICE** (*service, n parameters*)	
Description:	Executes PI service	
Parameters:	Service	PI service identifier
	n parameters	List of n parameters of PI Service. Individual parameters are separated by commas.

Example

```
PRESS (HS2)
  PI_SERVICE("_N_CREATO",55)
END_PRESS
PRESS(VS4)
  PI_SERVICE("_N_CRCEDN",17,3)
END_PRESS
```

Starting OEM services

The PI_START command executes a PI service based on OEM documentation.

Programming

Syntax:	PI_START("*Transfer string*")	
Description:	Executes PI service	
Parameters:	"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.

See also

List of PI services (Page A-17)

2.7.29 External functions (only HMI Advanced)

Description

Additional user-specific functions can be used by means of this call. External functions are stored in a DLL file and identified by an entry in the definition lines of the configuration file.

Note

An external function must have at least one return parameter.

Programming

Syntax:	**FCT***Function name = ("File"/Type of return/Types of permanent call parameters/Types of variable call parameters)*	
Description:	Declares additional, external functions	
Parameters:	Function name	Name of external function
	File	Complete path to DLL file
	Type of return	Data type of return value for function
	R, **I**, S, C, B.	Data types of **permanent** call parameters and return values. The data types are separated by commas.
	Variable or register	Data types of **variable** call parameters
Syntax:	**FCT***Function name (call parameter)*	
	Call parameter	List of all call parameters. Individual parameters are separated by commas.

Example

```
//M(SCREEN FORM1)
DEF VAR1 = (R)
DEF VAR2 = (I)
DEF RET = (I)
FCT InitConnection = ("c:\user\mydll.dll"/I/R,I,S/I,S)
                         ;  The external function "InitConnection" is declared. The
                            data type of what has been declared. The data type of
                            the return value is Integer, the data types of the
                            permanent call parameters are Real, Integer and String;
                            the data types of the variable call parameters are
                            Integer and String.
LOAD
  RET = InitConnection(VAR1+SIN(VAR3),13,"Servus",VAR2,
```

//M(SCREEN FORM1)

```
  REG[2])
                             ;  The value of the "InitConnection" external function is
                                assigned to RET along with the call parameters
                                VAR1+SIN(VAR3), 13, Servus, VAR2 and REG[2].
END_LOAD
```

Extract from DLL file

```
void __export WINAPI InitConnection(ExtFctStructPtr FctRet, ExtFctStructPtr
FctPar, char cNrFctPar)
FctRet->value.i   Return value of function
FctPar[0]->value.r   1st parameter (VAR1+SIN(VAR3))
FctPar[1]->value.i   2nd parameter (13)
FctPar[2]->value.s   3rd parameter ("Servus")
FctPar[4]->value.i   4th parameter (reference to VAR2)
FctPar[5]->value.s   5th parameter (reference to REG[2])
cNrFctPar   Number of parameters (5)
```

2.7.30 Programming example

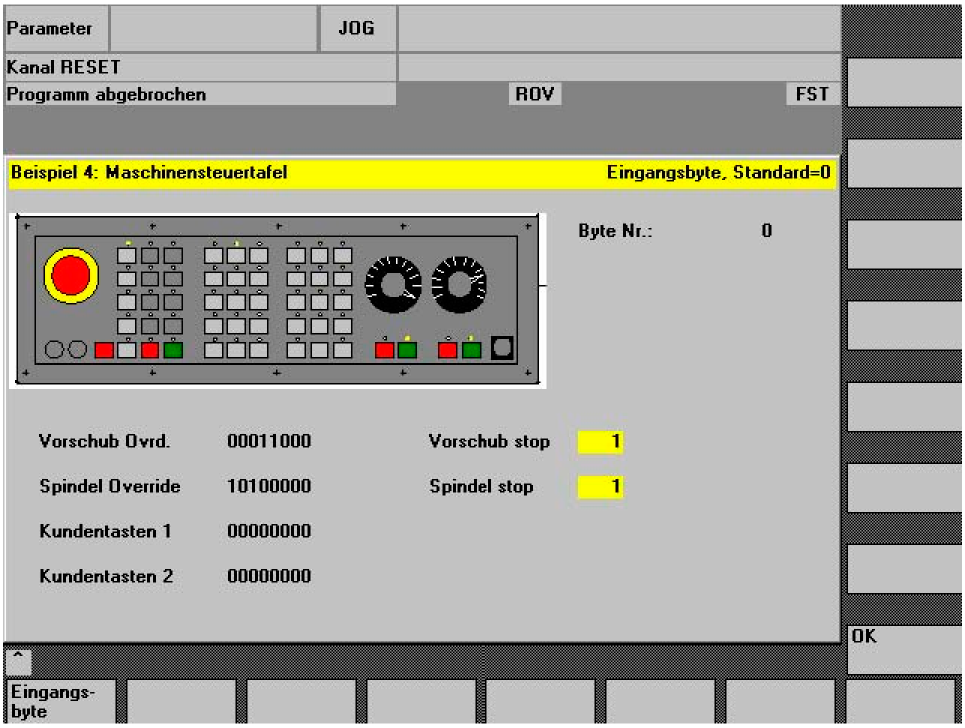
Programming

```
//S(Start)
HS7=("Example", se1, ac7)
PRESS(HS7)
LM("Screen form4")
END_PRESS
//END
//M(Screen form4/"Example 4: Machine control panel"/"MST.BMP")
DEF byte=(I/0/0/"Input byte, default=0","Byte
no.:",""/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"/280,180,100/380,180,50/0,11)
DEF spin=(IBB//0/"","Spindle
override",""/wr1//"EB0"/20,210,100/130,210,100),
spinstop=(B//0/"","Spindle
stop",""/wr1//"E2.4"/280,210,100/380,210,50/0,11)
DEF custom1=(IBB//0/"","User key
1",""/wr1//"EB6"/20,240,100/130,240,100)
DEF custom2=(IBB//0/"","User name
2",""/wr1//"EB7"/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=("")
VS8=("OK", SE1, AC7)

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

PRESS(VS8)
  EXIT
END_PRESS
```

Result



Beispiel 4: Maschinensteuertafel Eingangsbyte, Standard=0

Byte Nr.: 0

Vorschub Ovrd. 00011000 Vorschub stop 1

Spindel Override 10100000 Spindel stop 1

Kundentasten 1 00000000

Kundentasten 2 00000000

Figure 2-10 Machine control panel

Note

You will find further examples in the supplied tool box.

2.8 Operators

Overview

The following operators can be used when programming:

- Mathematical operators
- Relational operators
- Logic (Boolean) operators
- Bit operators
- Trigonometric functions

2.8.1 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

Note

The functions operate with radian measure. The functions SDEG() and SRAD() can be used for conversion.

Example: `VAR1.VAL = SQRT(2)`

Constants

Constants	
PI	3.14159265358979323846
FALSE	0
TRUE	1

Example: `VAR1.VAL = PI`

Relational operators

Relational 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 a command:	`IF`
	`...`
	`ENDIF`

Condition with two commands:	`IF`
	`...`
	`ELSE`
	`...`
	`ENDIF`

2.8.2 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 left by number of bits
                                    specified in VAR04. Then convert 32-bit value
                                    back to format of variable VAR02.
END_PRESS
```

Programming support 3

3.1 What does programming support do?

Overview

The purpose of the programming support system is to assist programmers in writing an NC program using the ASCII Editor. Customized user interfaces can also be created in the Programming support system. These interfaces are configured using the ASCII Editor and the tools provided by the "Expanding the Operator Interface" system.

The system provides the following standard tools for this purpose:

- Cycle support
- Free contour programming
- Contour definition programming
- Recompile
- Simulation

Note

For the sake of compatibility, the cycles support (//C...) function is still supported by the description language used in earlier versions and not by the syntax of the "Expanding the Operator Interface" system.

Creating new dialogs

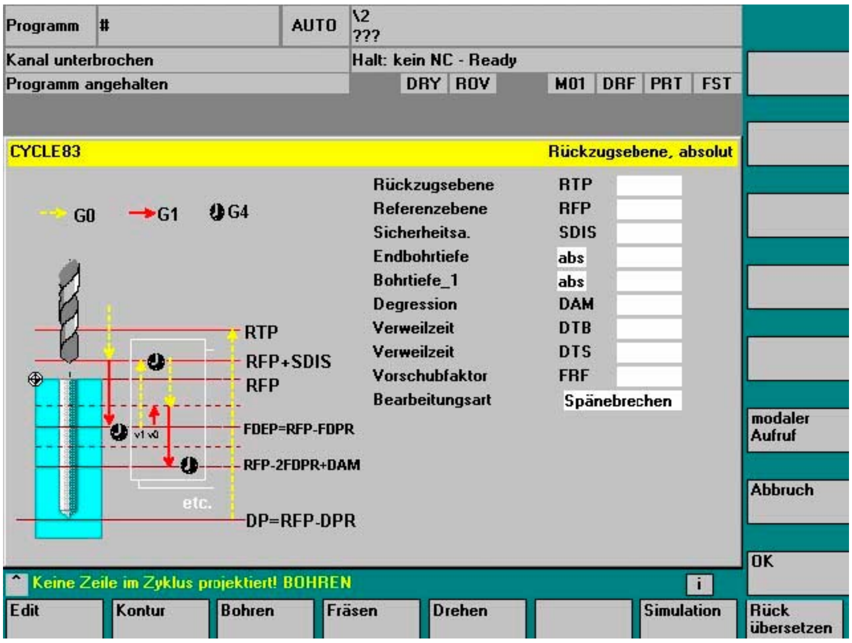
New operator interfaces are created with tools provided by the "Expanding the Operator Interface" system. However, there are some differences affecting programming support and these are described in this chapter.

Configuration file

The definition of new dialogs for programming support is stored in configuration file AEDITOR.COM.

- Newly configured dialogs can be displayed in the Editor main screen through selection of 5 start softkeys (horizontal softkeys 2, 3, 4, 5 and 6).
- The default labels for softkeys 2 to 5 are "Contour", "Drilling", "Milling" and "Turning".

- Horizontal softkeys 14 and 15 (softkeys 6 and 7 in the expanded menu) are assigned the "Gauge turning" and "Gauge milling" functions.



Programm # AUTO \2 ???

Kanal unterbrochen — Halt: kein NC - Ready

Programm angehalten

CYCLE83 — Rückzugsebene, absolut

Rückzugsebene	RTP	
Referenzebene	RFP	
Sicherheitsa.	SDIS	
Endbohrtiefe	abs	
Bohrtiefe_1	abs	
Degression	DAM	
Verweilzeit	DTB	
Verweilzeit	DTS	
Vorschubfaktor	FRF	
Bearbeitungsart	Spänebrechen	

Keine Zeile im Zyklus projektiert! BOHREN

Figure 3-1 View of the default setting

3.2 Cycle support

Use

When using cycle support in the parts programs, additional comment lines are generated in front of and behind the cycle call and used for recompilation. These lines start with ;# ... .

When cycles noted directly using the editor are called (also referred to as "old" cycles), then this information is missing. A description is provided in the following text as to how the information for cycles, contained in the lines in ;# can be provided by configuring INI files. For Siemens cycles, the required configuration files are included in the scope of supply.

This means that after recompilation it will be possible to work with the following cycle calls using the dialog support:

- Directly noted cycle calls
- Siemens cycle calls
- User cycle calls

Prerequisites

For these cycles, a configuration generated using the "Expand user interface" tools must exist or be generated.

For dialogs generated with the "Expand user interface" function for the purpose of supporting user cycles, the information in the dialog descriptive files can be used for this process.

Cycle-specific settings can be made in the configuration to define whether cycles with ;# lines are to be processed in the parts program or recompiled via the configuration files.

This process can be used to optionally generate the addition of ;# lines required for dialog support in the parts program for directly noted “old" cycles.

Configuration

The WIZARD.INI file can be stored in the following directories. The directories are browsed in this sequence:

..\user

..\oem

..\hmi_adv

..\mmc2

The following entries refer to cycle packages, e.g.:

```
INI_1=bohren.ini
INI_2=drehen.ini
...
INI_n=paket.ini
```

The information in the [MMC_CycleWizard] section is added so that it is up to date.

Cycle-specific entries in the following format are located in the relevant PAKET.INI files:

```
[cycleName]       ; Name of the cycle as a separate section
Mname=            ; Obligatory    dialog name
Dir=              ; Obligatory    Directory of file with dialog information
Dname=            ; Obligatory    File with dialog information
Output            ; Obligatory    Name of the OUTPUT block
Anzp=             ; Obligatory    Number of parameters (defined variables)
Version=          ; Optional      cycle version, without entry 0
Code_typ=         ; Optional      output with = 0 or without = 1 ;# lines
```

Example:

```
[CYCLE83]
Mname=CYC83
Dir=cst.dir
Output=bohren.ini
Anzp=17
Version=3
Code_typ=0
```

Different cycle versions

A differentiation is made between various versions of cycles with the same name using the version number. Main block/secondary block and block numbers in front of the cycle call are kept unchanged.

Cycle call line in the parts program:

```
/1234 :44 CYCLE94(, , )
```

After expansion by the string and ;# lines:

```
;NCG#CYC94#\CST.DIR\DREHEN.COM#NC1#1#*NCG;*RO*;*HD*
;##########*NCG;*RO*;*HD*
/1234 :44 CYCLE94(,,"",)
;#END#*NCG;*RO*;*HD*
```

See also

Recompile (Page 2-96)

Search Forward, Search Backward (SF, SB) (Page 2-98)

3.3 Activating a dialog from the NC program

Introduction

With HMI Advanced and HMI Embedded sl, dialogs defined by the user can be displayed. The appearance of the dialogs is defined through configuration (modification of COM file in the cycles directory).

The dialog is called and exited by function calls from the parts program. User-configured dialog screens do not modify the HMI system software (user interface). User-defined dialogs cannot be called simultaneously in different channels.

Command channel

The "Activate dialog from NC Program" function is also referred to as a "command channel".

Activating the command channel

One possible application of user-defined dialogs is, for example, to assign defined values to particular user variables (GUD) prior to a parts program run.

- Up to 2 channels:

  The "command channel" is activated by default for channels 1 and 2.

- More than 2 channels:

  For HMI Advanced, the "command channel" must be activated (if the Siemens measuring cycles have not yet been installed). This requires a modification to be made to the file F:\MMC2\ COMIC.NSK in the "Startup" operating area:

Select file **F:\MMC2\COMIC.NSK** in the "Startup" operating area using the softkeys "HMI" → "Editor" and attach the following text (after channels 1 and 2):

```
REM CHANNEL
TOPIC(machineswitch) COMIC_START(COMIC001MachineSwitch"...)
[compare the text for channels 1 and 2]
```

When the control is restarted (OFF/ON), the command channel is activated for the corresponding channel.

Activation with HMI Advanced

Content of the COMIC.NSK file:

```
REM ----- TYPICAL COMIC START
REM CHANNEL 1
TOPIC(machineswitch) COMIC_START("COMIC001MachineSwitch
",/Channel/Configuration/mmcCmd[u1],
/Channel/Configuration/mmcCmdQuit[u1])
REM CHANNEL 2
TOPIC(machineswitch) COMIC_START("COMIC002MachineSwitch
",/Channel/Configuration/mmcCmd[u2],
/Channel/Configuration/mmcCmdQuit[u2])
```

3.3.1 Structure of "MMC" instruction

Programming

Syntax:	**MMC** ("Operating area, command, Com file, dialog box name, user-data definition file, graphics file, display time or acknowledgment variable, text variables...", "Acknowledgment mode")	
Parameters:	Operating area	Name of softkey used to call the configured user dialogs. Default: CYCLES displayed as "Cycles" on softkey 14 and accessible via the <ETC> key.
	Command	PICTURE_ON Select screen PICTURE_**OFF** Deselect screen
	COM file	Name of the dialog screen file (max. 8 characters, in the user, manufacturer, or standard cycles directory). The dialog display appearance is defined here. The dialog screen is used to display user variables and/or comment texts.
	Dialog name	The individual dialogs are selected via the dialog names.

GUD file	User-data definition file accessed on reading/writing of variables.	
Graphics file (HMI Advanced only)	File name of the BMP graphic to be inserted	
Acknowledgment variable (HMI Advanced only)	Acknowledgment variable	in acknowledgment mode "A"
or display time	Display time	in acknowledgment mode "N"
Text variable	Screen header or comment text from a text variable in the COM file.	
Acknowledgment mode	"S" for **S**ynchronous	Acknowledgment via "OK" softkey
	"A" for **A**synchronous	Acknowledgment via configured softkeys
	"N" for **N**o Quit	No acknowledgment, but display time

Storage structure of graphics

*.bmp files are stored in resolution-specific subdirectories:

- For standard cycles:

\CST.DIR\HLP.DIR\640.DIR	For 640 dpi resolution
\CST.DIR\HLP.DIR\800.DIR	For 800 dpi resolution
\CST.DIR\HLP.DIR\1024.DIR	For 1024 dpi resolution

- For user cycles:

\CUS.DIR\HLP.DIR\640.DIR	For 640 dpi resolution
\CUS.DIR\HLP.DIR\800.DIR	For 800 dpi resolution
\CUS.DIR\HLP.DIR\1024.DIR	For 1024 dpi resolution

- For manufacturer cycles:

\CMA.DIR\HLP.DIR\640.DIR	For 640 dpi resolution
\CMA.DIR\HLP.DIR\800.DIR	For 800 dpi resolution
\CMA.DIR\HLP.DIR\1024.DIR	For 1024 dpi resolution

3.3.2 Example of MMC instruction

MMC instruction in the parts program

```
MMC("CYCLES,PICTURE_ON,T_SK.COM,PICTURE1,MGUD.DEF,PICTURE3.BMP,
TEST_1,A1","S")
```

CYCLES	Operating area
PICTURE_ON	Select dialog
T_SK.COM	File name in cycle directory
PICTURE1	Name of the dialog
MGUD.DEF	User data definition file
PICTURE3.BMP	Name of graphics file (HMI Advanced only)
TEST_1	Acknowledgment variable (HMI Advanced only) or display time in "N" mode
A1	Screen header or comment from a text variable (COM file)
S	Acknowledgment mode: Synchronized

User variables in definition directory

```
%_N_UGUD_DEF
;$PATH=/_N_DEF_DIR
DEF CHAN REAL TEST_1
```

CHAN	Applicable channel
REAL	Data type
TEST_1	Name of user variable

Dialog screen file in the cycles directory (*.COM)

```
//C3(Screen2)
R/ 15   75 / 5 /COMMENT, %1 %2 %3/ W,RJ / TEST_1 / ...
```

R	Variable Type: Real, Integer or String	
15 75	Permissible range: 15 to 75	
5	Default setting for user variable	
COMMENT, %1 %2 %3	Comment text with optional text variables	
W,RJ	Access type:	W = Read and write R = Read-only W, RJ = Read and write with comment J = Right-justified in relation to input/output field <without> = Left-justified in relation to input/output field
TEST_1	User variable	

Text variable

```
[Text variables]
A1 = Example 2: MMC instruction without acknowledgment
```

`A1`	Reference parameters for MMC instruction
`Example 2: MMC statement without acknowledgment`	Screen header or comment text

Note

For variable names, text variables and cycle names, capital letters must be used.

Configuring softkeys for dialog call

Softkey assignment for MMC command with asynchronous acknowledgment mode.

```
[PICTURE3]
SK1 = END
SK2 = Screen2
```

Softkeys SK0 to SK15 can be configured

3.3.3 Example 1: MMC instruction without acknowledgment

Selecting

```
N10 MMC("CYCLES,PICTURE_ON,T_SK.COM,PICTURE1,GUD4.DEF,,,A1","N")
N20 TEST_1 = 1
N25 G4 F10
N30 MMC("CYCLES,PICTURE_OFF","N")
M30
```

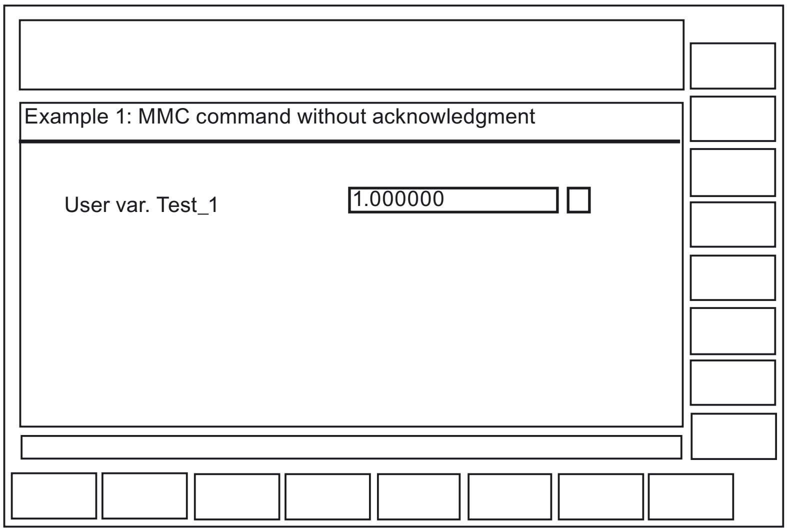
Parameters:

Dialog file (*.COM)

```
//C1(PICTURE1)
(R///USER VAR TEST_1/W/TEST_1///)
```

Text variable

```
[TEXT VARIABLES]
A1 =........Example 1: MMC command without
acknowledgment
```



Example 1: MMC command without acknowledgment

User var. Test_1 1.000000 □

Figure 3-2 Example 1 Without acknowledgment

Sequence of operations

The user variable TEST_1 from GUD4.DEF with header A1 is displayed for a short time. The hold time comes from parts program block N25.

3.3.4 Example 2: Hold time and optional text variables

Selecting

```
N10
MMC("CYCLES,PICTURE_ON,T_SK.COM,PICTURE6,GUD4.DEF,,10,T1,G1","N")
N15 G4 F15
N30 MMC("CYCLES,PICTURE_OFF","N")
M30
```

Parameters:

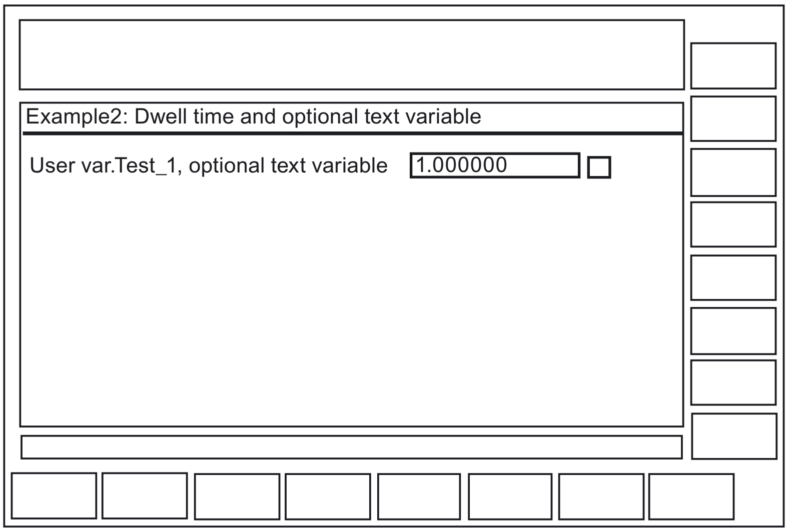
Dialog file (*.COM)

```
//C6(screen6)
(R///USER VAR TEST_1,%1/W/TEST_1///)
```

Text variable

```
[TEXT VARIABLES]
T1 = Example 2: Dwell time and optional text
variables ...
G1 = Optional text variable
```

The 7th parameter is interpreted as the display time for the mode without acknowledgment (10 seconds). The contents of the table are then deleted. The dialog remains on the screen until PICTURE_OFF is selected. The 8th parameter (T1) is the text variable for the header. If there is no entry, the operating area name, "cycles", is displayed. The parameters 9 to 23 are optional text variables ("G1=optional text variable"). In this COM file, the optional text variables must be preassigned in the [Text variables] section.


Example2: Dwell time and optional text variable

User var.Test_1, optional text variable 1.000000 □

Figure 3-3 Example 2 Dwell time

Sequence of operations

The comment text from the COM file (USER VAR TEST_1) is expanded in this example at the position of the first wild card (%1) to include the contents of the text variable

"G1=optional text variable". By calling the text variables contained in the MMC command (9th to 23rd parameters), messages or names can be "composed" in this way.

3.3.5 Example 3: MMC instruction with synchronous acknowledgment mode

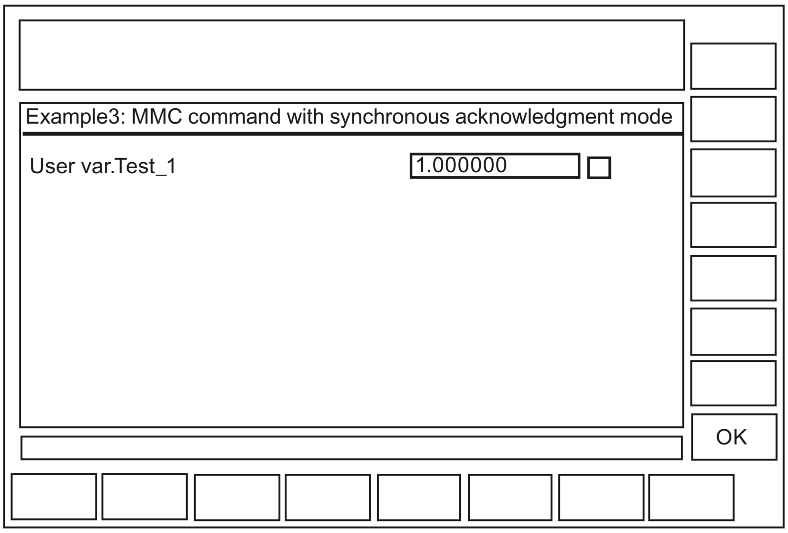
Selecting

```
N15 MMC("CYCLES,PICTURE_ON,T_SK.COM,PICTURE1,GUD4.DEF,,,F1","S")
N18 STOPRE
N20 TEST_1 = 5
N25 MMC("CYCLES,PICTURE_OFF","N")
M30
```

Parameters:

Dialog file (*.COM)	`//C1(PICTURE1)`
	`(R///USER VAR TEST_1/W/TEST_1///)`

Text variable	`F1 = ...Example 3: MMC command with synchronous acknowledgment mode...`



Example3: MMC command with synchronous acknowledgment mode

User var.Test_1 1.000000 □

Figure 3-4 Example 3 Synchronous acknowledgment mode

Sequence of operations

The user variable Test_1 is displayed until the "OK" softkey is pressed. The user variable has also been overwritten with the value 5 here.

Without STOPRE, this assignment would be carried out **before** the keyboard input (after it with STOPRE).

3.3.6 Example 4: Positioning of input/output field

Description

By specifying position parameters in the COM file, you can insert the comment field, or input and output field, at any point in the display area.

Selecting

```
N15 MMC("CYCLES,PICTURE_ON, T_SK.COM; Screen2,GUD4.DEF,,,C1","S")
N20 TEST_3 = 5
N30 MMC("CYCLES,PICTURE_OFF","N")
N40 M30
```

Parameters:

The two parameters each consist of three numerical values; these specify the position and length of the field. The values are specified in Twips, with 15 Twips roughly corresponding to one pixel. The field height is defined as being 250 Twips.

Dialog file (*.COM)

```
//C2(Screen12)
(R///Var.Name/R/TEST_3/6000,2800,8000/200,3000,750
0)
```

`/6000,2800,8000`		Position of the comment field
`/200,3000,7500`		Position of input/output field
First value = 0	→	Automatic positioning to default setting
No details	→	Positioning as on PCU 20 (default setting)

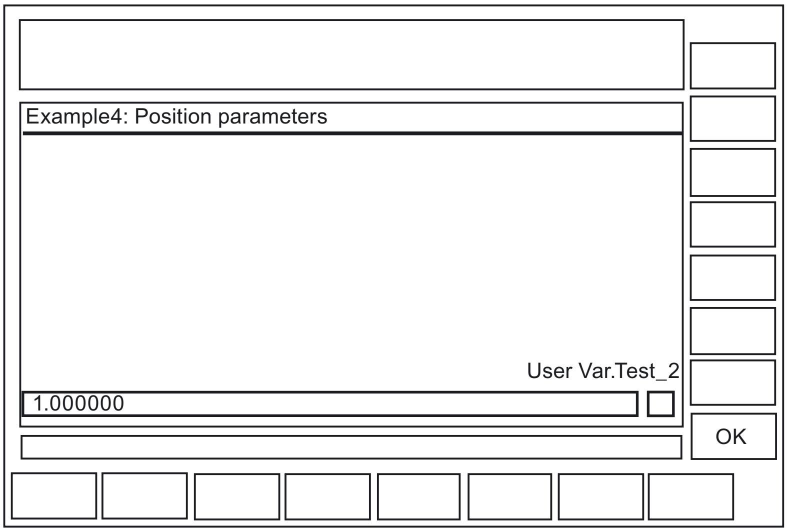
Meaning of values:

`(./6000,2800,8000/....)`

`6000`	Distance from left border
`2800`	Distance from upper border
`8000`	Field length

Sequence of operations

It is possible to graphically configure 16 comment and 16 input or output fields. Where there are more than 16 fields, they can be controlled via a scroll bar.



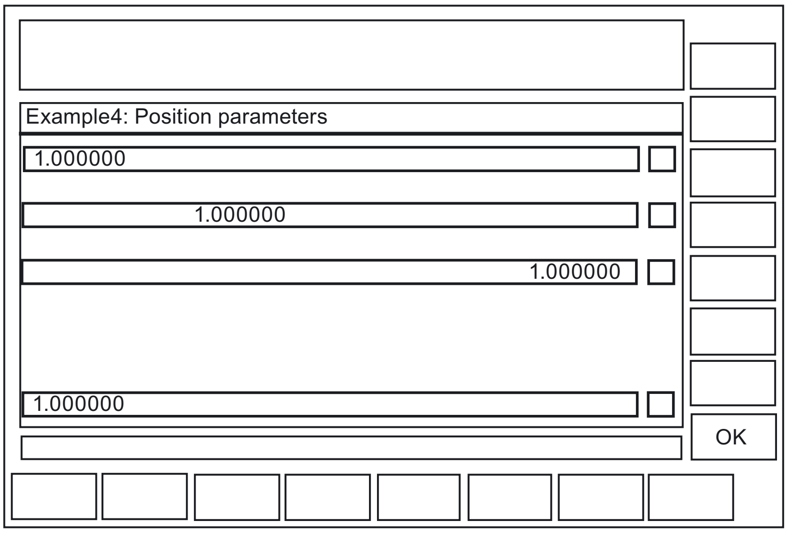
Example4: Position parameters

User Var.Test_2

1.000000 □

Figure 3-5 Example 4a Position parameters

To ensure that the cursor control works seamlessly, the configured fields must overlap:



Example4: Position parameters

1.000000 □

1.000000 □

1.000000 □

1.000000 □

Figure 3-6 Example 4b Position parameters

3.3.7 Example 5: Displaying graphics in the dialog screen

Description

Graphics created with Paintbrush, for example, can be displayed in the dialog by means of a graphics file specification.

A comment text for the graphic can also be preassigned using the COM file. You can position this comment text by specifying position parameters.

Note

You can only move the graphic itself by repositioning it in the graphics program.

Selecting

```
N10
MMC("CYCLES,PICTURE_ON,T_SK.COM,PICTURE8,GUD4.DEF,GRA.BMP,,M1","S")
N20 MMC("CYCLES,PICTURE_OFF","N")
N30 M30
```

Parameters:

Dialog file (*.COM)	`//C8(SCREEN8)`
	`(I///Insert picture no. 2///4000,3000,7500)`
	`(I///Created with Paintbrush ///4000,3250,7500)`

Text variable	`M1 = ......Example5: Insert picture.......` Pictures are created, e.g., with the "Paintbrush" program. Picture size: 300X500 pixels, you can only change the picture size with the graphics program.

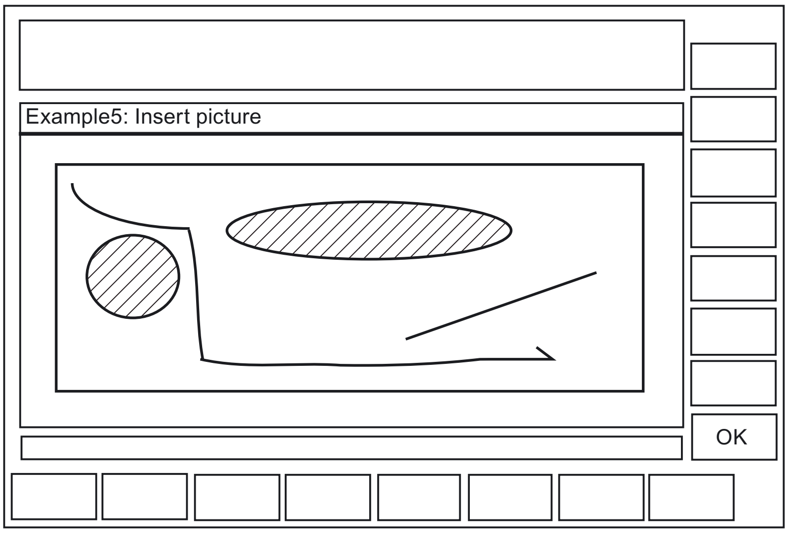


Figure 3-7 Example 5 with graphic

3.3.8 Example 6: Displaying OPI variables

Description

You can display OPI variables.

Selecting

```
MMC("CYCLES,PICTURE_ON,T_SK.COM,SCREEN7,GUD4.DEF,,TEST_1,J1","S")
```

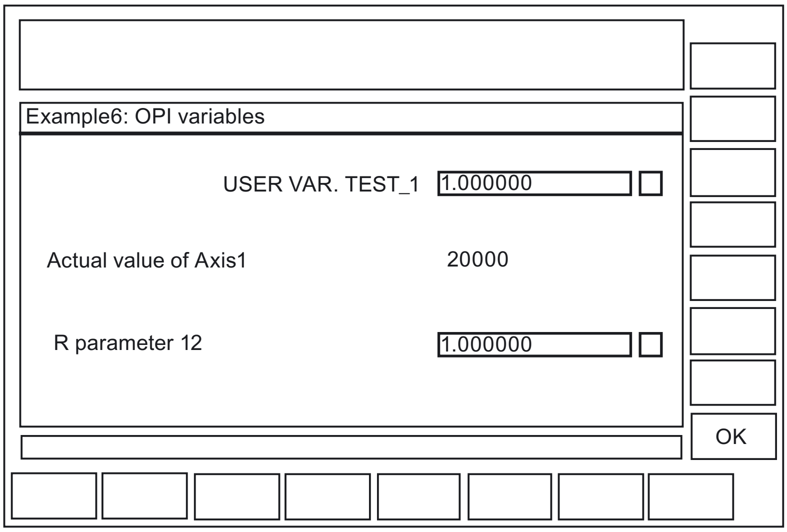
Parameters:

Dialog file (*.COM)

```
//C7(Screen7)
(R///Test_1/R/Test_1)
(I///); (Interpreted as space line))
(R///Actual value from axis 1/R/$actual value)
(R//1/R parameter 12/W/$R[12])
```

Text variable

```
J1 = ...Example7: OPI variables

[OPIVar]
$Actual
value=/Channel/machineaxis/actToolbasePos[u1,1]
$R[12]=/Channel/Parameter/rpa[u1,12]
```


Example6: OPI variables

USER VAR. TEST_1 1.000000 □

Actual value of Axis1 20000

R parameter 12 1.000000 □

Figure 3-8 Example 6 with OPI variable

Sequence of operations

A space line is created under the variable TEST_1.

The actual axis value is read-only.

R12 is preset with 1.

3.3.9 Example 7: Asynchronous acknowledgment mode with softkeys

Description

Softkeys can be preprogrammed in the COM file for display in asynchronous mode, linked with the acknowledgment variable and evaluated in the parts program.

Selecting

```
N10 QUIT_1 = "START"
N20
MMC("CYCLES,PICTURE_ON,T_SK.COM,PICTURE3,GUD4.DEF,"QUIT_1,K1","A")
N30 LABEL0:
N40 STOPRE
N50 IF MATCH (QUIT_1,"SK1") >= 0 GOTOF LABEL1
N60 IF MATCH (QUIT_1,"SK2") >= 0 GOTOF LABEL2
N70 GOTOB LABEL0
N80 LABEL2:
N90 MMC("CYCLES,PICTURE_ON,T_SK.COM,PICTURE1,GUD4.DEF,"N1","N")
N100 G4F10
N110 LABEL1:
N120 MMC("CYCLES,PICTURE_OFF","N")
N130 M30
```

Parameters:

Dialog file (*.COM)

```
//C3(Screen3)
(S///USER VAR QUIT_1/W/QUIT_1//)
```

Text variable

```
[TEXT VARIABLES]
K1 = ..Example8: MMC command with asynchronous
acknowledgment mode
N1 = ..Example8: Picture2
```

Softkey

```
[SCREEN3]
SK1 = END
SK2 = Screen2
```

Program structure

The acknowledgment variable is defined as a string.

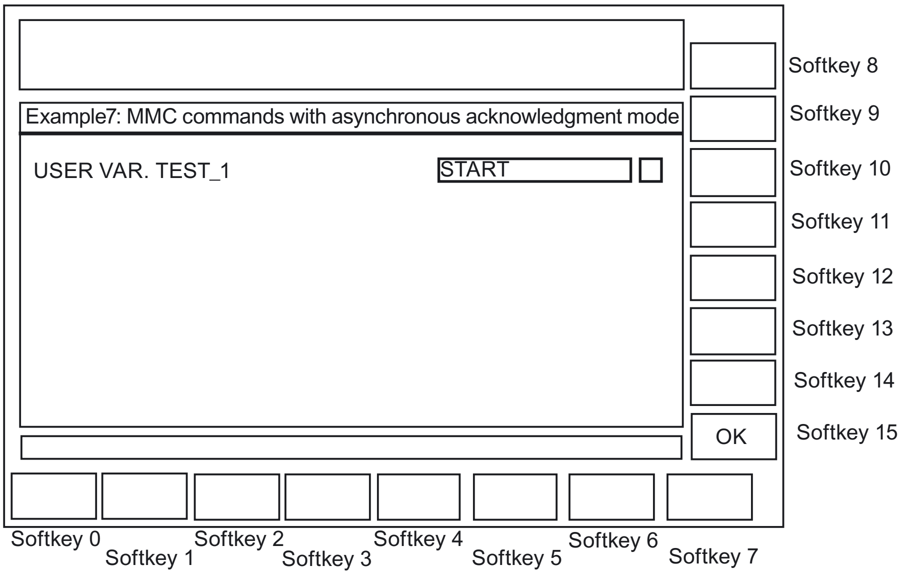
```
Length of the string: >= 20
;(Values < 20 are evaluated internally only,□ SK0 ... SK15 are
entered at position 17...20 when a softkey is actuated).
```

The string is assigned a value in the parts program and any old softkey information is deleted.

Before the parts program can branch as a function of the acknowledgment variable, the block search must be halted by the STOPRE command.

```
IF Match (Quit_1,"SK1") >= 0 GotoF Label1
;searches for a string within the string.
```

If no softkey has been pressed, the loop is executed again.



Example7: MMC commands with asynchronous acknowledgment mode

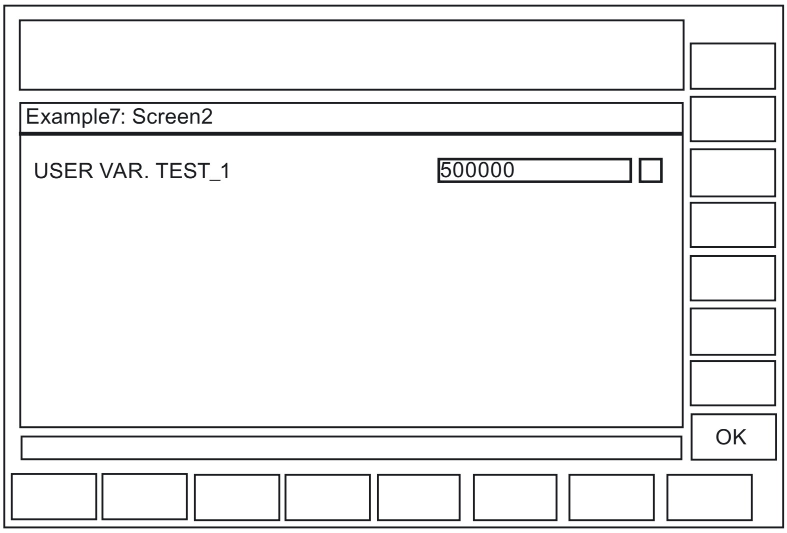
USER VAR. TEST_1 START □

Figure 3-9 Example 7: Asynchronous acknowledgment mode

Sequence of operations

The screen called via the asynchronous MMC command continues to be displayed until one of the two configured softkeys is actuated:

- With the "END" softkey, the user dialog is immediately exited.
- With the "Picture2" softkey, a further dialog is then displayed for 10 s.



Example7: Screen2

USER VAR. TEST_1 500000 □

Figure 3-10 Example 7: Fig. 2:

Configure hotkeys and PLC keys 4

4.1 Introduction

Overview

This chapter describes the configuration of the following control elements:

- 6 hotkeys on OP 010, OP 010C and SINUMERIK keyboards with hotkey block, as well as the <MACHINE> and <MENU SELECT> keys, whose assignment can be changed.
- Keys evaluated by the PLC, e.g., keys on the machine control panel
- Events that are evaluated by the PLC as PLC keys or "virtual keys", and that can trigger configured operating sequences in the HMI program.

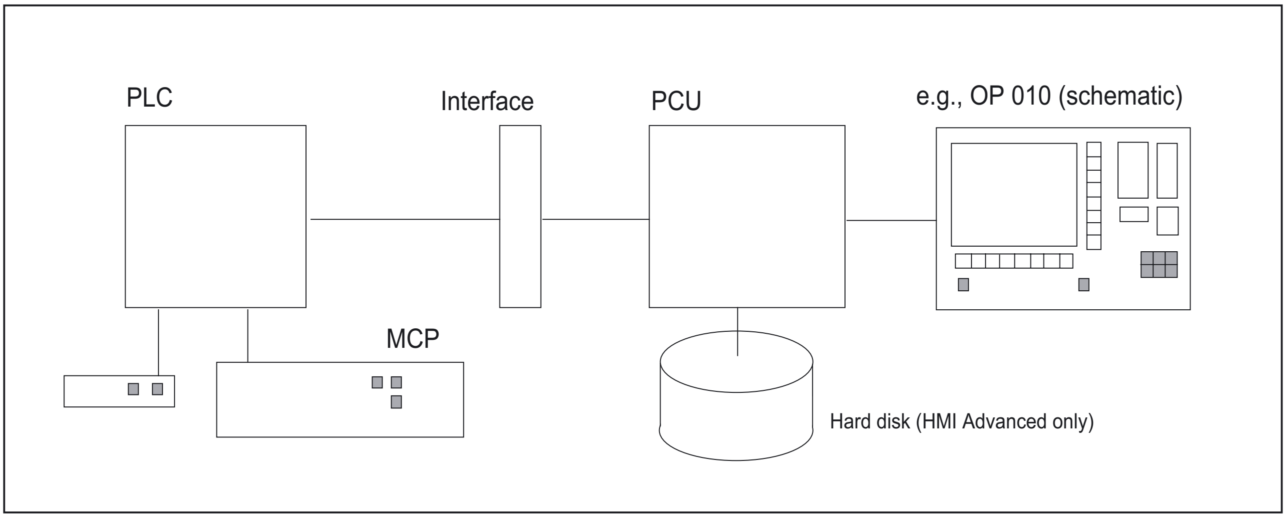


Figure 4-1 Configuration with OP 010

Application

The hotkeys and keys can be used for, e.g., the following tasks:

- Selection of operating areas (e.g., Machine, Parameters, etc.)
- Selection of specific submenus (e.g., selection of the alarm screen in the Diagnosis operating area)
- Initiation of actions (e.g., selection of the tool list in the Parameters area and pressing of softkey HS3)
- Specific selection of menus created using the "Expanding the Operator Interface" system

- Selection of additional screens, depending on the current operating situation, in screens created using the "Expanding the Operator Interface" system.

Configuration

Configuration is carried out using the "Expand user interface" functions.

- The 6 OP hotkeys can be used to display any HMI operating area directly. This shortens the usual selection process via the main menu. This changes the default assignment of the 6 hotkeys.
- The interface between the PLC and the HMI enables a PLC key to be transferred from the PLC to the HMI. The operations initiated on the HMI system in this way can be configured. Key numbers 50 to 254 are available for use by the PLC.
- As an option, the <Machine> and <MENU SELECT> keys can be configured in the same way as the 6 OP hotkeys and are assigned to HK7 and HK8.

4.1.1 OP hotkeys

Assignment (default setting)

The 6 OP hotkeys are laid out in 2 lines, each of which contains 3 keys:

Line 1 Label (no symbol)		Configured as
OP-specific:		
OP 10	Machine	HK1
OP 10C	Machine	HK1
OP 10S	Position	HK1
Program		HK2
Offset		HK3

Line 2 Label (no symbol)	Configured as
Program Manager	HK4
Alarm	HK5
Custom	HK6

Optional HK7 and HK8:

The <Machine> and <MENU SELECT> keys can be configured in the same way as HK1 to HK6. This makes it possible to disable the default settings of these keys and to activate new, user-defined functions.

Label (no symbol)	Configured as
Machine	HK7
MENU SELECT	HK8

You can find more information about HK7 and HK8 in the chapter "Configuring <M> key and <MENU SELECT> as HK7, HK8".

Note

Hotkeys 1 and 7 (<M> key) cannot be distinguished by the hardware on the OP 10S ("Position" labeling). Hot key 7 is always triggered when one of the keys is pressed. If HK1 is configured, this event can only be triggered by an external (MF2) keyboard.

Hotkey assignment on the MF2 keyboard

Hotkey	OP label	Key on MF2
HK1	Position	<SHIFT+F11>
HK2	Program	<END> (NB)*
HK3	Offset	<Page Down> (NB)*
HK4	Prg.Manager	<Home> (NB)*
HK5	Alarm	<Page Up> (NB)*
HK6	Custom	<SHIFT+F12> or Cursor Down (NB)*
HK7	M Machine	<SHIFT+F10>
HK8	Menu Select	<F10>

*) Key is located on numeric keypad: <NumLock> must be off.

4.1.2 Functions of the keys on delivery

Delivery condition

On delivery of the system, the assignment of hotkeys to functions is entered in the KEYS.INI file.

With HMI Advanced, the file can exist in various directories:

- user
- oem
- add_on *)
- mmc0w32 *)
- mmc2 *)
- hmi_adv *)

*) These directories are reserved for Siemens.

The files are executed in the above sequence. Entries in the directory, which appears at the beginning of the sequence will disable existing entries in directories, which appear later in the sequence.

On delivery, the assignments for HK1 to HK6 are stored in KEYS.INI in the **mmc2** directory for the standard system:

Key		Function
HK1	Position	Machine operating area, last screen
HK2	Program	Program operating area, last screen
HK3	Offset	Parameters operating area, last screen
HK4	Prg. manager	Main screen programming
HK5	Alarm	Diagnosis operating area, alarm screen
HK6	Custom	Default Custom operating area, last screen (user interface configured by the user)

The assignment for ShopMill/ShopTurn applications is as follows:

Key		Function
HK1	Position	Machine operating area, last screen
HK2	Program	Program editor, last status
HK3	Offset	Offset area, last status
HK4	Prg. manager	Program directory, last status
HK5	Alarm	Diagnosis operating area, alarm screen
HK6	Custom	Default Custom operating area, last screen (user interface configured by the user)

4.2 Configuration

4.2.1 Configuration overview

Overview

Below is a schematic diagram illustrating the associations between configured OP hotkeys and PLC keys and the "Expand user interface" configuration:

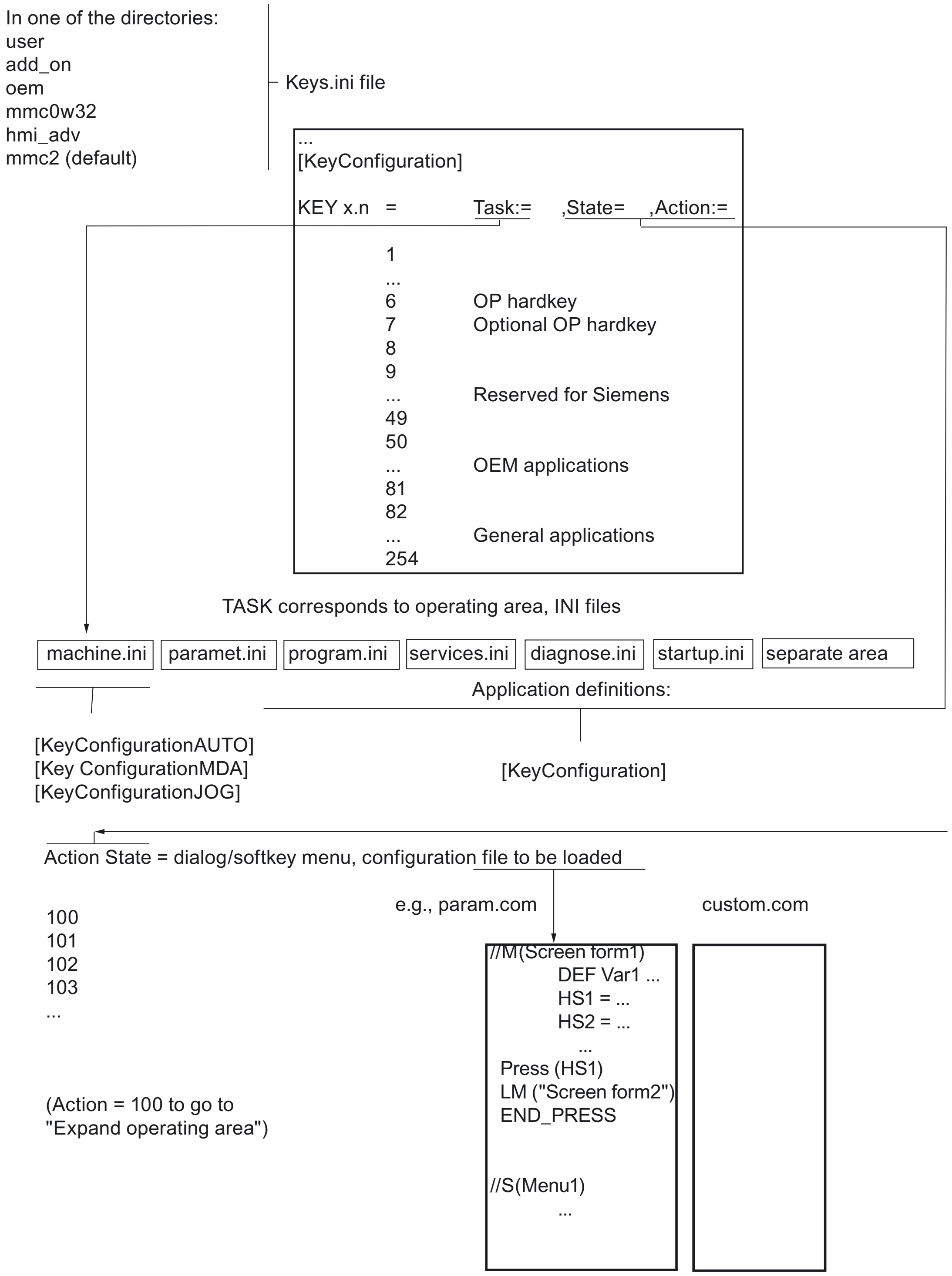


Figure 4-2 Configuration overview

4.2.2 Configuration in the KEYS.INI file

Overview

Every instance of the KEYS.INI configuration file in the above directories drives the the hotkey responses in a section [KeyConfiguration]. Every entry (line) defines the response of the system to a specific actuation (in the sense of multiple actuation of specific hotkeys). These are referred to as hotkey events below.

KEYS.INI configuration file

The KEYS.INI configuration file has a dedicated section for assigning the ini files for PLC keys. Without this entry, the ini files will not be detected.

The default settings are as follows:

```
[HMI_INI_FILES]
Task0 = machine.ini
Task1 = paramet.ini
Task2 = program.ini
Task3 = services.ini
Task4 = diagnose.ini
Task5 = startup.ini
Task6 = shopmill.ini
```

(Alternatively: Task6 = shopturn.ini depending on the system, mandatory!)

```
Task11=custom.ini
```

Note

A power On is required in order for the changes to KEYS.INI to take effect.

Entry format

Details about the input format can be found under "Hotkey event". To better understand the repeated actuation of the same hotkey used in that subsection, multiple keypress will be described first.

Multiple keypress

A function extension determines the sequence when the hotkey is pressed repeatedly: The number "n" in the key data in file KEYS.INI represents the number of keypress repeats, thus allowing a separate task/state/action combination to be assigned each time the hotkey is pressed. This means that each time the key is pressed, a change of state can take place, and a screen and a softkey can be selected.

When the user changes to another operating area, the repeat status of the hotkey is cleared. In the new area, the hotkey is considered not to have been pressed. It does not matter how the user changed to the other operating area (operating area menu, etc.). In addition, each time a function key is pressed (e.g., softkey, area switchover key, channel switchover key, etc.), the status is reset.

Inputting data into fields in the current dialog does not interrupt the repetition sequence. Pressing a softkey in the current dialog interrupts the repetition sequence with the softkey that was used to call the current dialog.

Note

Multiple keypress is possible for HK1 - HK8 and for keys HK9 - HK49, which are reserved for Siemens.

Example of multiple hotkey keypress

The first time the hotkey is pressed, the associated area is activated and a state/action may be initiated in this area (entry 1). Pressing the hotkey again will execute the subsequent entries for this hotkey and an area switchover will not take place. Once all the entries configured for a hotkey have been executed, the cycle is repeated. The entries are always executed in ascending order (from 0-9). Execution in descending order is not possible. There must be no "event gaps" in the configuration (missing entries). A gap is treated like the end of the chain and processing will restart at Key x.0 the next time a key is pressed.

Configuration:

```
[KeyConfiguration]
KEY1.0 = Task:=0, State:=10,       ; Hotkey 1, 1st activation
Action:=2
KEY1.1 = State:=10, Action:=3      ; Hotkey 1, 2nd activation
KEY1.2 = Action:=4                 ; Hotkey 1, 3rd activation
```

Entry in the KEYS.INI file for selecting the machine operating area:

```
KEY1.0 = Task:=0                   ; in the last screen the first
                                     time the key is pressed
KEY1.1 = Task:=0, State:=0         ; in the start screen the second
                                     time the key is pressed
KEY1.2 = State:=0, Action:=3       ; in the start screen the third
                                     time the key is pressed and
                                     selection of the 3rd softkey
KEY1.3 = State:=0, Action:=100       in the start screen the 4th
                                     time the key is pressed and
                                     selection of "Expand user
                                     interface" function with the
                                     configured action ≤ 100 in
                                     MACHINE.INI
```

Assignment of the softkeys to the action values

Horizontal softkeys (SKHi)	Action
SKH1	0
SKH2	1
SKH3	2
SKH4	3
SKH5	4
SKH6	5
SKH7	6
SKH8	7

Vertical softkeys (SKVi)	Action
SKV1	8
SKV2	9
SKV3	10
SKV4	11
SKV5	12
SKV6	13
SKV7	14
SKV8	15

Key	Action
<RECALL>	16
<ETC> (HMI Embedded sl only)	17

4.2.3 Programming a hotkey event

Description

The maximum number of attributes, which can be assigned to a hotkey event, is 4. The KEY key attribute identifies the event and must always come first. All other attributes are optional, but at least one other attribute must be specified. These attributes can appear in any sequence in the line.

Programming a hotkey event

Syntax: KEYx.n = Task:=task, State:=state, Action:=action

Parameters:

Key (HotKey)

The value x contains the hotkey number **x** and a discrete event **n** corresponding to the nth activation of the key (when the key is pressed several times in succession).

The range of values of the hotkey number goes from 1 to 254.

The range of values for the event n goes from 0 to 9 (10 keystrokes)

Hotkeys 1 to 8 are assigned to the keys of the OPs. Hotkeys 9 to 49 are reserved for Siemens. PLC-specific keys from 50 to 254 are also available. These keys are not actual hotkeys, but are used for selecting screens via the PLC (virtual keys). Keys 50 to 254 can also be configured without a task. In this case, the event is always assigned to the current task.

Task (Operating area)

A hotkey is assigned to an operating area (task) via a task number. The task number also defines the horizontal softkey to which the operating area is assigned in the area switchover menu. This creates a permanent connection between the task number and the corresponding softkey. In the HMI system, there are three softkey menus (two in HMI Embedded sl) reserved for the area switchover function.

The default assignments are as follows:

Pressing the area switchover key selects softkey menu 1.

SK1 → Task0 ...

SK8 → Task7

Pressing the <ETC> key again accesses softkey menu 2

SK1 → Task8 ...

SK8 → Task15

Pressing the <ETC> key again accesses softkey menu 3 (only possible in HMI Advanced):

SK1 → Task16 ...

SK8 → Task23

Pressing the <ETC> key again returns to softkey menu 1. In HMI Advanced, the range of values for the task numbers is between 0 and 23, in HMI Embedded it is between 0 and 15. If no task is specified, the event is assigned to the current task (current operating area).

The task numbers can be accessed:

- From the REGIE.INI file for HMI Advanced.
- With the SCK (Software Configuration Kit) tool in HMI Embedded sl.

An application-specific assignment between SKs and operating areas (tasks) can be preset during configuration:

- HMI Advanced:
  [TaskConfiguration] section in the REGIE.INI file
- HMI Embedded sl: SCK "Modify configuration"

State

The State attribute can be used to select a specific screen within a task. The range of values depends on the individual application and is limited to between 0 and 65534 (65535 is used by the system).

Action

The Action attribute can be used to select a specific softkey within a task. In order to be effective, the attribute must be set from a defined state, e.g., the main menu.

The range of values depends on the corresponding application, with a maximum of between 0 and 17. A special role is played by the Action attribute with a value ≥ 100, which initiates a screen selection via "Expand user interface".

Actions in the value range 0 - 99 are not executed for ShopMill/ShopTurn.

See also:

Chapter "List of selectable states"

4.2.4 Expansions and special cases

Configuring the M key and Menu Select as HK7, HK8

The <M> key (Machine) and <MENU SELECT> key can be optionally configured as hotkeys. Doing so causes these keys to lose their original definition and function. The new functionality is defined by the KEYS.INI configuration file. If there is a definition in the configuration file for the key, but no response was saved, then the key does not have a function.

Hotkey assignment:

		MF2 keyboard	Hotkey
M MACHINE	Key <Machine>	SHIFT+F10	HK7
MENU SELECT	Key <MENU SELECT>	F10	HK8

If there is no entry in the keys.ini configuration file for hotkey 7 or hotkey 8, the key will not be mapped as a hotkey but will retain its original function (compatible mode). A key definition without a response will be assigned the <empty> attribute.

Example:

```
[KeyConfiguration]
Key7.0 = Task:= 3, State := 10  ; Activate mapping of the <M> key
                                  to hotkey  7 and define new□ key
                                  reaction
                                ; Activate mapping of the <MENU
                                  SELECT> key (F10) to □hotkey 8□;
                                  key has no function
Key8.0 = <empty>                ; No reaction assigned
```

Expanded functions in HMI Advanced

The machine manufacturer can overload the entries in KEYS.INI in directory mmc2 with custom settings. These settings can be entered in the /user or /oem directories searched earlier during execution. Only deviations in the settings must be stored, not the complete block.

Special cases for Action in HMI Advanced:

>

The <ETC> key, Action 17, has no effect.

Special cases in HMI Embedded sl

Special cases Task, State, and Action:

- The range of values for the Task is limited to between 0 and 15.
- HMI Embedded sl always selects the last screen combination that was active before the task change (state not configured).
- The range of values for State is limited to 0. Only the main menu of the operating area can be selected.
- The range of values for Action is restricted to between 0 and 17 for:
  - Horizontal SK 1 to 8
  - Vertical SK 1 to 8
  - Recall
  - ETC

4.2.5 Expanded functions for PLC keys

Overview

Changes of task and state can also be configured for PLC keys.

Range for Action:	
For standard applications	0 – 17
For "Expand user interface" applications	≥100

The PLC-specific keys have numbers between 50 and 254, with numbers 50 to 81 reserved for OEM applications.

Examples:

- The configuration shown here does not execute a change of task or state and, therefore, remains in the current task and screen. An Action 100 is forwarded.

```
[KeyConfiguration]
HK50.0 = Action:=100
```

- A change of task and state could be configured here in exactly the same way.

```
HK50.0 = Task:=1, State:=10, Action:=100
```

4.3 PLC interface

4.3.1 Structure of the interface

Overview

In the PLC interface, area DB19. Byte10 is provided for key selection. Here, the PLC can directly select **one** key between 50 and 254.

(Keys 1 to 49 are reserved for Siemens, keys 50 to 254 are specific PLC keys.)

Note

In M:N mode, the area for the 2nd HMI interface is DB19.Byte60.

Acknowledgment

An HMI system acknowledgment is divided into two steps:

- In the first step, the HMI software transfers the control information 255 to DB19.Byte10.
- In the second step, the actual acknowledgment takes place on the PLC in which DB19.Byte10 is cleared.

This is necessary in order that, despite the lack of synchronism between the HMI and the PLC, the same key code transferred twice in immediate succession can be detected by the HMI. This defined virtual key activation enables the HMI to identify every PLC key sequence uniquely. The control information is of no significance to the PLC program (transparent) and must not be changed.

Next key input

If the transfer byte is set to 0, the PLC program can specify a new key. The current key request is processed in parallel in the HMI system. The request leads to a switchover to the corresponding task or initiates a state/action command in the current task.

If the task switchover cannot be executed, the operator interface will send a message to indicate this.

Requirements of the PLC program

A new request can only be set if the HMI system has acknowledged the previous request (0 in the interface). If the PLC program is deriving the key from a machine control panel key or from another source, it must provide sufficient intermediate storage for the key to ensure that no data is lost if the key is pressed very quickly.

Selecting dialogs from PLC

There is an interface between the PLC and HMI Embedded sl for selecting dialogs. These PLC dialogs offer the same means of display and functions as for the dialogs selected via the softkeys.

4.3.2 Description of the PLC display selection

Description of the interface

The interface contains the display number, control bits from the PLC to the HMI and control bits from the HMI to the PLC. The interface requires a total of 8 bytes in DB19, 4 bytes for each HMI Embedded sl.

Because each NCU can be addressed by up to two HMI Embedded sls at the same time in systems with "multiple operator panel fronts and multiple NCUs", this interface can exist in duplicate.

References: /FB2/, "Multiple operator panel fronts and multiple NCUs" (B3)

Structure of the interface

The interface between HMI Embedded sl and the PLC uses the following data:

HMI 1:	DB19.DBW28:	Screen number
	DB19.DBB30:	Control bits PLC → HMI, PLC byte
	DB19.DBB31:	Control bits HMI → PLC, HMI byte
HMI 2:	DB19.DBW28:	Screen number
	DB19.DBB80:	Control bits PLC → HMI, PLC byte
	DB19.DBB81:	Control bits HMI → PLC, HMI byte

PLC-Byte	Bit0	Display selection
	Bit1	Display selection
HMI byte	Bit0	Display selection or display deselection accepted
	Bit1	Display is being selected or deselected
	Bit2	Display is selected
	Bit3	Display is deselected
	Bit4	Error, display selection not possible
	Bit7	Inactive bit

Two bytes are used for the **display number** to be transferred by the PLC; these are one PLC byte and one HMI byte for **coordinating** the display selection.

Operating the interface

The interface is operated by the PLC program supplied by the manufacturer, using the following functions (DB 19 extract, first HMI interface):

	PLC → HMI	Selection	Deselection	HMI → PLC	Selection	Deselection
DBW 28	Screen number	(1)				
Bit	DBB 30			DBB 31		
0	Display selection	1 (2) 0 (4)		Select/deselect accepted	1 (3) 0 (6)	1 (2) 0 (3)
1	Display deselection		1 (1) 0 (4)	Display is being selected Display is being deselected	0 (3) 1 (5)	0 (3)
2				Display is selected	0 (3) 1 (7)	0 (3)
3				Display is deselected	0 (3)	0 (2) 1 (3)
4				Error, display selection not possible	0	0 (2)

	PLC → HMI	Selection	Deselection	HMI → PLC	Selection	Deselection
5				–		
6				–		
7				Inactive	0	

Display selection

The relations with the individual steps are shown in the above table by the numbers in brackets (step numbers).

- The PLC enters the display number in the number word.
- The PLC sets bit0 in the PLC byte for display selection if bit0 and bit7 are set to zero in the HMI byte.
- HMI Embedded sl acknowledges to the PLC that it has accepted the display selection by setting bit0 in the HMI byte. At the same time, the value zero is written to bit3 and bit4 in the HMI byte.
- The PLC resets bit0 in the PLC byte.
- HMI Embedded sl acknowledges to the PLC that the interpreter has started by setting bit1 in the HMI byte.
- The HMI resets bit0 in the HMI Embedded sl byte.
- Bit2 is set in the HMI byte if the corresponding display appears.

Display deselection

The relations with the individual steps are shown in the above table by the numbers in brackets (step numbers).

- The PLC sets bit1 for display deselection in the PLC byte if bit1 and bit2 are set in the HMI byte and bit7 is reset.
- HMI Embedded sl acknowledges to the PLC that it has accepted the display deselection by setting bit0 in the HMI byte. At the same time, the value zero is written to bit3 and bit4 in the HMI byte.
- HMI Embedded sl acknowledges the display deselection to the PLC by setting bit3 and resetting bit0, bit1 and bit2 in the HMI byte if the interpreter is terminated.
- The PLC resets bit1 in the PLC byte.
- After display deselection by the PLC or the EXIT command, HMI Embedded sl switches back to the last selected HMI display.

Error when selecting a display

Bit4 is set in the HMI byte if the interpreter has not responded after 20 seconds or if the required display cannot be opened.

The states of the HMI that do not support PLC display selection are indicated to the PLC via bit7 in the HMI byte; e.g., in HMI Embedded sl during switchover between standard and ShopMill/ShopTurn interface.

Operating areas

With **HMI Advanced**, the displays are shown in a separate operating area similar to the measuring cycle displays. This operating area can be selected manually by means of the sixth softkey (PLC displays) on the second operating area menu. If no PLC display is active when the area is selected, the text "Currently no PLC display active" is displayed in the header. The PLC is informed when the area is selected/deselected manually via bit2 of the HMI byte.

It is also possible to change to other operating areas via the MMC 103's operating area after a display has been selected by the PLC.

With **HMI Embedded sl**, it is not possible to manually select/deselect displays selected by the PLC via an operating area. The PLC display remains in the foreground even after a change of operating area has occurred.

4.3.3 Configuring the dialog selection

Description

The interface is activated by means of the section [PLC_SELECT] in the COMMON.COM file during HMI boot. The display number is assigned to the configured display in the section [PLC_SELECT] of the COMMON.COM file.

Configuration

Syntax:	**PC***i* = *Display name, date, comment*	
Description:	Assigns display number to configured display	
Parameters:	i	Number of the display in the interface
	Display name	Dialog identifier
	File	File in which the dialog is configured
	Comment	Comment on dialog

Example

```
[PLC_SELECT]
PC1= CYC82, drilling.com        ;   Assignment line
PC2= CYCLE90, millthre.com
PC3= ...
```

See also

Search function for COMMON.COM (Page 6-13)

4.4 Selecting dialog/softkey menus

4.4.1 Assigning INI files to operating areas

Overview

An Action value ≥ 100 automatically switches the HMI software to "Expand user interface" operational sequences.

If Action ≥ 100, the configuration of the **INI files** determines which softkey menu or screen form should be displayed and in which state.

HMI Advanced/HMI Embedded sl

The following INI files and sections can be configured for the jump entry to "Expand user interface":

Operating area	File	Section
Machine	machine.ini	[KeyConfigurationAuto] [KeyConfigurationMDA] [KeyConfigurationJOG]
Parameter	paramet.ini	[KeyConfiguration]
Program	program.ini	[KeyConfiguration]
Services	services.ini	[KeyConfiguration]
Diagnostics	diagnosis.ini	[KeyConfiguration]
Start-up	startup.ini	[KeyConfiguration]
Custom	custom.ini	[KeyConfiguration]

ShopMill on NCU/ ShopTurn on NCU

In the case of ShopMill and ShopTurn on NCU, the files are accordingly named SHOPMILL.INI and SHOPTURN.INI. The following sections, in which "Expand user interface" functions can be configured, are parts of these files:

```
[MachineManual]
...
[MachineAutomatic]
...
[ProgramManager]
...
[Program]
...
[MessagesAlarms]
...
[ToolsZeroOffset]
```

```
...
[MachineMDI]
```

4.4.2 Configuring "Expand user interface" functions

Description

One or more states (according to the State list) can be configured for each action (≥ 100) and you can also specify which "Expand user interface" function is to be initiated. If nothing is configured in these INI files, no screen forms or softkey menus will appear.

Programming Action.State

Syntax:	Action.State = Dialog/softkey menu to be loaded, configuration file	
Parameters:	Action	The action is Action ≥ = 100 configured in KEYS.INI.
	State	State in which the application is presently in
	Configuration file	File in which the configuration is stored.
	LS/LM	"Expand user interface" commands for loading softkeys/dialogs

Examples

```
[KeyConfiguration]
100.10=LS("Softkey1","param.com")
100.30=LM("Form1","param.com")
101.10=LS("Softkey1","param.com")
101.30=LM("Form2","param.com")
102=LM("Form2","param1.com")
```

Explanations:

- For the first line, for example, this means:

  If an Action 100 is initiated when in state (status/screen) 10, the softkey menu configured in the param.com file will be displayed with the name Softkey1.

- For the last line, for example, this means:

  If Action 102 is initiated, the dialog with the name Screen form2 will be called in the current dialog. This dialog is configured in the param1.com file.

4.5 List of selectable states

4.5.1 Selectable states in HMI Advanced

Overview

In general, the following conditions apply to all tasks in HMI Advanced:

No configuration:	Maintains current state.
Configuration with 0:	The basic state of an operating area is activated.

"Machine" operating area

In the Machine operating area, the status is always dependent on the status of the machine (AUTO, MDA, JOG, REF). The following can be selected via the PLC key:

- The main screen for the relevant mode

State	BAG	Machine function	Display
0	JOG	REF	JOG/REF main screen
0	JOG	none	JOG main screen
0	JOG	REPOS	REPOS main screen
0	MDI	none	MDA start screen
0	MDI	TEACH	MDA/Teach main screen
0	MDI	REF	MDA/REF main screen
0	AUTO	none	Auto main screen

- The main actual value screens for each mode (softkey vertical 6)

State	BAG	Machine function	Display
10	JOG	none	Zoom actual values JOG
20	MDI	none	Zoom actual values MDA
30	AUTO	none	Zoom actual values Auto

- Possibly still in JOG / MDA

  Switching between the WCS and MCS is possible via the PLC.

State	BAG	Machine function	Display
60	JOG	none	Handwheel selection JOG
70	JOG	none	Increment selection JOG
80	MDI	none	Handwheel selection MDA

"Parameters" operating area

It is not possible to move from the current screen in this area.

"Program" operating area

State	Function	
10	Data selection	
20	Program management	
70	Log	

Services operating area

State	Function	
10	Data in	
20	Data out	
40	Manage data	
60	Data selection	
80	External drives	
90	Series startup	
100	upgrading	

Diagnostics operating area

The diagnostics main screen is the alarm overview.

The following horizontal softkeys can be used to access other states from this state:

State	Function	
10	Alarms	
20	Alarms	
30	Alarm log	
40	Service displays	
50	PLC status	

Messages, alarm log and service displays can always be accessed.

Startup operating area

The startup main screen is an overview of the available NC axes and drives.

State	Function	
0	NC axes and drives	
10	Machine data	
40	PLC status	

State	Function	
50	Optimization/test	(V7.1 and later)
60	HMI	

Custom operating area

See chapter "Custom operating area".

4.5.2 Selectable states in HMI Embedded sl

Overview

There are the following configuration options for the state in HMI Embedded sl:

A configuration:	Maintains current state.
Configuration with 0:	The basic state of an operating area is activated.

4.5.3 Selectable states in ShopMill on NCU

Machine Manual

Legend:

*	If option available (display MD is set)
Custom screen	These designated functions can be configured with the "Expanding the Operator Interface" system. If this type of configuration is available, it is activated. Otherwise, the standard ShopMill screen will appear.

State	Function
19	Main screen
2	T, S, M, etc.
30	Workpiece zero
5	Workpiece zero - set up edge
7	Workpiece zero – /user screen form
31	Workpiece zero – align edge/user screen form
32	Workpiece zero – spacing 2 edges/user screen form
33	Workpiece zero – rectangular edge
8	Workpiece zero – arbitrary edge/user screen form
34	Workpiece zero – rectangular pocket
9	Workpiece zero – 1 hole/user screen form

State	Function
35	Workpiece zero – 2 holes
36	Workpiece zero – 3 holes
37	Workpiece zero – 4 holes
38	Workpiece zero – rectangular spigot
10	Workpiece zero – 1 circular spigot /user screen form
39	Workpiece zero – 2 circular spigots
40	Workpiece zero – 3 circular spigots
41	Workpiece zero – 4 circular spigots
42	Workpiece zero - set up plane
11	Workpiece zero – calibration probe, length*/user screen form
12	Workpiece zero – calibration probe, radius*
50	Measure tool
16	Tool gauging – length manual/user screen form
17	Tool gauging – diameter/user screen form
13	Tool gauging – length auto*/user screen form
14	Tool gauging – diameter auto*/user screen form
51	Measure tool – /user screen form
15	Measure tool – calibration probe*/user screen form
52	Measure tool – calibration fixed point*/user screen form
60	Swiveling*
4	Position
18	Face milling
1	ShopMill settings
90	– /user screen form

MDI

State	Function
20	MDI

Machine Auto

State	Function
200	Main screen
210	Program control
220	Block search
230	– /user screen form
242	Simultaneous recording – top view*
243	Simultaneous recording – 3-plane view*
244	Simultaneous recording – volume model*
250	Settings

Program Manager

State	Function
300	NC directory
310	Parts programs*
320	Subprograms*
330	User directory 1 *
340	User directory 2 *
350	User directory 3 *
360	User directory 4 *
380	Standard cycles*
381	Manufacturer cycles*
382	User cycles*
383	User directory 5 *
384	User directory 6 *
385	User directory 7 *
386	User directory 8 *

Program

State	Function
400	Machining plan / G code editor
412	Simulation – top view*
413	Simulation – 3-plane view*
414	Simulation – volume model*

Messages/alarms

State	Function
500	Alarms
510	– /user screen form
520	– /user screen form

Tools/zero offsets

State	Function
600	Tool list
610	Tool wear
620	User tools list*
630	Magazine
640	Zero point shift
650	R parameters

State	Function
660	– /user screen form
680	User data
690	Machine data

4.5.4 Selectable states in ShopTurn on NCU

Machine manual (without "Machine manual" option)

Legend:

*	If option available (display MD is set)
Custom screen	These designated functions can be configured with the "Expanding the Operator Interface" system. If this type of configuration is available, it is activated. Otherwise, the standard ShopMill screen will appear.

State	Function
19	Main screen
2	T, S, M, etc.
30	Workpiece zero
31	Workpiece zero – /user screen form
34	Workpiece zero – /user screen form
35	Workpiece zero – /user screen form
36	Workpiece zero – /user screen form
37	Workpiece zero – /user screen form
38	Workpiece zero – /user screen form
40	Workpiece zero – /user screen form
5	Workpiece zero – measure edge Z
50	Measure tool
51	Measure tool – length manual X/user screen form
52	Measure tool – length manual Z/user screen form
53	Measure tool – zoom*/user screen form
54	Measure tool – /user screen form
55	Measure tool – /user screen form
56	Measure tool – calibration probe*/user screen form
57	Measure tool – /user screen form
58	Measure tool – automatic Z *
59	Measure tool – automatic X *
4	Position
18	Face milling*
80	Cutting*

State	Function
90	– /user screen form (tailstock)
1	ShopTurn settings

Machine hand (with "Machine manual" option)

State	Function
19	Main screen
50	Measure tool
51	Measure tool – length manual X/user screen form
52	Measure tool – length manual Z/user screen form
53	Measure tool – zoom*/user screen form
54	Measure tool – /user screen form
55	Measure tool – /user screen form
56	Measure tool – calibration probe*/user screen form
57	Measure tool – /user screen form
58	Measure tool – automatic Z *
59	Measure tool – automatic X *
1300	Straight
1400	Bore
1410	Drilling – centered
1420	Drilling – thread centered
1433	Drilling – centering*
1434	Drilling – drilling*
1435	Drilling – reaming*
1440	Drilling – deep-hole drilling*
1453	Drilling – tapping*
1454	Drilling – thread milling*
1500	Turning
1513	Turning – cutting 1
1514	Turning – cutting 2
1515	Turning – cutting 3
1523	Turning – groove 1
1524	Turning – groove 2
1525	Turning – groove 3
1533	Turning – undercut form E
1534	Turning – undercut form F
1535	Turning – undercut thread DIN
1536	Turning – undercut thread DIN
1543	Turning – thread, longitudinal
1544	Turning – thread, taper
1545	Turning – thread, facing
1550	Turning – cut-off

State	Function
1600	Milling*
1613	Milling – rectangular pocket*
1614	Milling – circular pocket*
1623	Milling – rectangular spigot*
1624	Milling – circular spigot*
1633	Milling – longitudinal groove*
1634	Milling – circumferential groove*
1640	Milling – polyhedron*
1670	Milling – engraving*
1730	Simulation – 3-window view*
1740	Simulation – side view*
1750	Simulation – front view*
90	– /user screen form (tailstock)
1	ShopTurn settings

MDI

State	Function
20	MDI

Machine Auto

State	Function
200	Current block display
210	Program control
220	Block search
230	– /user screen form
242	Simultaneous recording – 3-window view*
243	Simultaneous recording – side view*
244	Simultaneous recording – front view*
250	Settings

Program Manager

State	Function
300	NC directory
310	Parts programs*
320	Subprograms*
330	User directory 1 *
340	User directory 2 *
350	User directory 3 *

State	Function
360	User directory 4 *
380	Standard cycles*
381	Manufacturer cycles*
382	User cycles*
383	User directory 5 *
384	User directory 6 *
385	User directory 7 *
386	User directory 8 *

Program

State	Function
400	Machining plan / G code editor
412	Simulation – 3-window view*
413	Simulation – side view*
414	Simulation – front view*

Messages/alarms

State	Function
500	Alarms
510	– /user screen form
520	– /user screen form

Tools/zero offsets

State	Function
600	Tool list
610	Tool wear
620	OEM tool list*
630	Magazine
640	Zero point shift
650	R parameters
660	– /user screen form
670	Spindles
680	User data
690	Machine data

"Custom" operating area 5

5.1 Delivery condition and use

Overview

The operating areas described thus far can be extended and modified using "Expanding the Operator Interface" tools. Extensions can only be applied to softkeys, which have not been used previously.

Using the tools described below, it is possible to configure a separate operating area in both HMI Embedded sl and HMI Advanced, where all 8 horizontal and 8 vertical softkeys are available for a user-specific user interface.

The default name of this operating area is "Custom".

On OPs with hotkey block (e.g., OP 010, OP 010C), the "Custom" operating area can be directly selected with:

- "Custom" hotkey
- Softkey 4 in the 1st horizontal expanded menu (default setting)

Condition on delivery

The "Custom" operating area displays an empty window with a configurable header across the whole area of the local menu. The "Custom" operating area also permits a configurable text in the operating area display field in the global menu.

All softkeys are empty and can be assigned as required by the customer using "Expand user interface".

When switching from and back to the "Custom" area, the screen active before exiting the "Custom" operating area is active.

- HMI Advanced

  The "Custom" operating area is available by default on delivery and can be reset via an entry in the REGIE.INI file or assigned to any other horizontal softkey.

- HMI Embedded sl

  The "Custom" operating area is available by default. It is supplied with the application diskette and can be assigned to any horizontal softkey by the customer using the Software Configuration Kit (SCK.exe).

5.2 Activating the operating area

HMI Advanced

The operating area is enabled in the REGIE.INI file and is assigned to a softkey.

Example

Activation via horizontal softkey 4 in the 1st expanded menu (default setting).

- The entry must be made in the [Miscellaneous] section:

```
[TaskConfiguration]
Task11 = name := custom, Timeout := 12000
```

- In the REGIE.INI file, this operating area can also be selected as the standard startup area. The entry must be made in the [Miscellaneous] section:

```
[Miscellaneous]
PoweronTaskIndex = 11
```

The corresponding lines can be deactivated by adding a ";" comment character at the beginning of the line if the "Custom" operating area is not to be selected or should not be activated on startup. The area can be assigned to a different softkey by changing the task number.

HMI Embedded sl

Using the display machine data MD 9016: MM_SWITCH_TO_AREA, you can define in which operating area HMI Embedded sl should start up (the "Custom" operating area can be specified here).

The value in the MD provides the number of the softkey for the required operating area.

Default:	12
Horizontal softkeys 1 to 8:	1 – 8
Softkeys in the expanded menu:	9 – 16

The HMI software for the "Custom" application evaluates the CUSTOM.INI file and decides whether the area should be displayed. In HMI Embedded sl, there is a section [Activate] for this:

```
[Activate]
Activate=True
```

The operating area is assigned to a horizontal softkey by the customer using the Software Configuration Kit (SCK.EXE), menu option "Modify configuration".

In order for the Custom operating area to be activated by default on start-up, the softkey numbers must be entered in display machine data MD 9016: MM_SWITCH_TO_AREA.

If this display MD has the value -1, HMI Embedded sl will start up in the operating area specified as the start-up area with SCK. On delivery, softkey 1 on the main menu is in the "Machine" operating area.

Activation by keys

- Hotkey

  The "Custom" hotkey on the OP 010 and OP 010C always activates this operating area (default setting). Other additional configurations are possible.

- Horizontal softkey

  Activation via the horizontal softkey 4 in the 1st expanded menu is the default setting. The softkey assignment can be changed in HMI Advanced; in HMI Embedded sl, this is done with the help of the Software Configuration Kit (SCK).

Behavior during operating area change

When you change from the "Custom" operating area to another operating area and back again, the window that was active when you closed the "Custom" operating area reappears the next time you open it.

5.3 Defining the start dialog

Overview

The inputs in the CUSTOM.INI file are used to define the start dialog and those in the RE_xx.INI file for the operating area name in HMI Advanced.

- Dialog header

  A text with the dialog header can be entered in this section [Header]. This can be in the form of either text or an alarm text number, which will make the header language-dependent:

  ```
  [Header]
  Text="XY special functions"
  Text=$80XXX
  ```

  Default setting: `Text = "Custom"`

- Picture in start dialog

  In the [Picture] section, it is possible to enter a path to a picture that is displayed at the start of the application:

  ```
  [Picture]
  Picture=\directory\bild.bmp
  ```

- Operating area name

HMI Advanced	HMI Embedded sl

The name specified appears on the top left of the start dialog displayed.
`[HSoftkeyTexts]`
`HSK11 = "Custom"`
	The name specified appears on the top left of the start dialog displayed.
	`[Task name]`
	`Notation: Text = $80xxx`

- Softkey labels

HMI Advanced	HMI Embedded sl

The name entered as the operating area name in the RE_xx.INI file is also displayed on the configured softkey.
"xx" stands for the language code.
	The text entered is displayed on the softkey assigned. If nothing has been entered, "Custom" will appear as the default setting.
	`[Softkey]`
	`Text = $80xxx`

All other elements in the "Custom" operating area, such as softkey menus or input/output fields and the relevant functions, must be configured using the "Expand user interface" tools.

In the "Custom" application **all** softkeys are available as "Expand user interface" softkeys.

These softkeys are configured in the CUSTOM.COM file (like the standard version of AEDITOR.COM included in the scope of supply) as described below.

In HMI Embedded sl, the COMMON.COM file for customer start softkeys must contain references to CUSTOM.COM. The COMMON.COM file is supplied so that all "Custom" operating area softkeys are linked to CUSTOM.COM.

Configuring environment 6

6.1 Scope of supply

Overview

The software that interprets the configuration files for user interfaces and provides or activates the functions described is included in the scope of delivery of the HMI software, along with an ASCII editor for creating the content of the configuration files (program editor).

Products

ShopMill on NCU and ShopTurn on NCU are based on HMI Embedded sl. HMI Embedded WIN32 and HMI Advanced can be combined on a single hardware platform. Thus, the operating systems of these products are available as alternatives to HMI Advanced on a single hardware platform.

Differences in the system

Because of the differences in the hardware, the configuration files are created in another way:

- HMI Advanced on PCU 50 has a hard disk.
- HMI Embedded sl only has RAM and user memory on the CF card.
- HMI Embedded WIN32 on PCU 50 or PC with Windows evaluates hard disk configuration files but operates in the same way as HMI Embedded sl in all other respects.

Creating displays

If the configured screen forms are to include images, then a suitable graphics program (e.g., MS Paint) will be required.

See also

Search function principle (Page 6-10)

6.2 Creating configuration data

6.2.1 Using the COMMON.COM file

Overview

When using HMI Advanced, no entries are required in the COMMON.COM file.

In HMI Embedded sl, this central control file contains the following information:

- Assignment between start softkeys and configuration files
- Assignment of display numbers to configuration files in PLC interface DB19
- Control entries (size of LOG file, available memory on the temporary directory for configuration files).

Default setting for HMI Advanced/HMI Embedded sl:

Main screen	Horizontal softkey	Configuration file
Machine JOG	1	MA_JOG.COM
Machine MDA	1	MA_MDA.COM
Machine Automatic	2	MA_AUTO.COM
Parameter	7	PARAM.COM
Program	8	PROG.COM
Services	7	SERVICE.COM
Diagnostics	7	DIAG.COM
Start-up expanded softkey menu	7 6, 7	STARTUP.COM
Editor (reserved)	2, 3, 4, 5	AEDITOR.COM
Editor expanded softkey menu	6 6, 7	AEDITOR.COM

Naming convention and file size

- HMI Embedded sl

  All file names must follow DOS conventions (xxxxxxxx.com).

  The total permissible number of configuration files is 10.

  Color format for the Help displays: 256-color bitmaps in bmp format.

  The memory requirement for the displays and configuration files is determined solely by the size of the CF card used. It is not possible to state what the maximum possible number of files can be.

- HMI Advanced

  Files with the names of the start softkeys are searched for in directories in the order specified above. If files of the same name are stored in different directories, then the first file detected according to the search strategy is interpreted. The sizes of configuration files are not subject to any particular restrictions. It should be noted, however, that large files are processed more slowly.

6.2.2 Structure of the COMMON.COM file

Overview

The COMMON.COM file is supplied together with the cycles. It contains various sections for specific hardware settings. For HMI Embedded sl, section [MMC_DOS] is relevant for the "Expand user interface" system.

Configuring COMMON.COM

Syntax	Parameter = *Value*	;	Any number of blanks can be placed between Parameter, the '=' sign and *Value*.
Description	[MMC_DOS]	;	Start of the DOS section
		;	Text contained within a line after a semicolon ; is treated as a comment and is not evaluated

Parameter: All parameters are optional.

SCxxx=*File*	Softkey connection: Connection between softkey and configuration file "xxx" represents an internal softkey identity of the start softkey. The softkey identity must be entered directly after SC. Softkeys will not be displayed unless a softkey connection is defined for them.
HCyyy = *File*	Hotkey connection: Connection between hotkey and configuration file. "yyy" represents an internal hotkey identity of the start hotkey. The hotkey identity must be entered directly after HC. Only hotkeys for which a hotkey connection has been defined have an effect.
File	The configuration file, which contains the softkey and dialog definitions. File names may not exceed a length of 8 characters. The file extension is added with a dot. Example SC101= my_file.com ; (my_file.com on HMI, NC)

Note

Changes to COMMON.COM only take effect following a restart.

Control entries

Syntax: CHK_FILE_EXIST=*ram*

Identifiers: This control flag: specifies whether configuration files must be copied each time by the NC or whether a check must be made as to whether the files are already stored on the RAM drive of the HMI.

Parameters: ram Possible values:

0: A check whether the file already exists on the temporary drive is not carried out. This mode is only set while the configuration files are being set up online on the NC. Changes to configuration files in the NC then become operative immediately on the HMI Embedded sl, slowing down the display building process.

1: Default setting (if CHK_FILE_EXIST has not been set): The configuration files are read once into the temporary directory and executed thereafter from there. This means Runtime is improved, but there is no reaction to changes in the configuration files in the NC.

Syntax: LOGSIZE=*kB*

Identifiers: A LOG file named ERROR.COM is created on the temporary directory of the HMI; its size is determined by this parameter setting.

Parameters: KB Size of the LOG file in kilobytes (max. size = 64 KB).

Syntax: RAMDISK_SIZE = *kBrd*

Identifiers: RAMDISK_SIZE can be located at any position - within or outside the sections. If the element occurs several times, then the **first** occurrence is relevant.

If, **after** copying, the selected size has been exceeded, all of the COM files in the temporary directory are deleted **before** the next copying operation. (Files for dialogs in the background for operating area change are retained.)

Parameters: kBrd Size of available RAMDISK in KB.

Default: 300 KB
(if the RAMDISK_SIZE element was not specified)

Example of COMMON.COM

```
[MMC_DOS]
sc101=tooth.com                  ; MACHINE
sc111=mda.com                    ; The files are stored on the flash memory of
                                   the HMI.
sc122=auto.com
sc207=param.com                  ; PARAMETER
sc314=aeditor.com                ; PROGRAM, Editor
sc315=aeditor.com
sc316=aeditor.com
sc407=dienste.com                ; SERVICES
sc507=diagnose.com               ; DIAGNOSIS
sc607=inbetrn.com                ; STARTUP
sc826=cmm.com                    ; ShopMill, Machine, AUTO
sc857=cmm.com                    ; Messages / alarms
sc858=cmm.com
sc867=cmm.com                    ; Tools, WO
CHK_COMMON.COM=1                 ; Rapid execution of the HMI
LOGSIZE=30                       ; Size of the error log (LOG file) 30 KB

[PLC_SELECT]                     ; Notification of displays, which can be called
                                   by the PLC
PC1= CYC82, drilling.com         ; Fig. 1:
PC2= CYCLE90, millthre.com       ; Fig. 2:
```

6.2.3 Configuring start softkeys

Overview

The start softkeys named here can be used to activate the associated configuration files. The possible start softkeys for dialogs are predefined. Additional start softkeys are not possible. Start softkeys are specific to operating areas.

Programming

Syntax	SCxxx = *File*	
Identifier	Softkey connection: Connection between softkey and configuration file "xxx" represents an internal softkey identity of the start softkey.	
Parameter	*File*	Name of configuration file

Entry points in the operating areas

Operating area	SCxxx	Output dialog	
MACHINE	SC101	MACHINE JOG	Horizontal SK 1
	SC111	MACHINE MDA	Horizontal SK 1
	SC122	MACHINE AUTO	Horizontal SK 2
PARAMETER	SC207	Main screen of PARAMETERS	Horizontal SK 7
PROGRAM	SC308	Main screen of PROGRAM	Horizontal SK 8
	SC312	1. softkey line of the editor	Horizontal SK 2
	SC313	1. softkey line of the editor	Horizontal SK 3
	SC314	1. softkey line of the editor	Horizontal SK 4
	SC315	1. softkey line of the editor	Horizontal SK 5
	SC316	1. softkey line of the editor	Horizontal SK 6
	SC326	2. Softkey menu of the editor	Horizontal SK 6
	SC327	2. Softkey menu of the editor	Horizontal SK 7
SERVICES	SC407	Main screen of SERVICES	Horizontal SK 7
DIAGNOSIS	SC507	Main screen of DIAGNOSIS	Horizontal SK 7
IBN	SC607	Main screen of START-UP	Horizontal SK 7
	SC616	2. Softkey menu of the editor	Horizontal SK 6
	SC617	2. Softkey menu of the editor	Horizontal SK 7

The file names listed in chapter "Table of start softkeys" are already stored as default settings. However, the matching files must be created on the NC or HMI by the user.

See also

List of start softkeys (Page A-1)

6.2.4 Language-dependent text

Overview

The language-dependent texts for dialogs are stored in ASCII text files. The syntax is the same as that of an alarm text file .

Language-dependent texts can be used for:

- Softkey labels
- Headings
- Help texts
- Any other texts

Permissible file names

The file names are defined as follows:

Alsc.txt	Contains the language-dependent texts for the Siemens standard cycles
Almc.txt	Contains the language-dependent texts for the manufacturer cycles
Aluc.txt	Language-dependent user texts

Format of a text entry

Syntax	8xxxx 0 0 "Text"		
Description	Assignment between text number and text in the file		
Parameter	xxxx	5000 to 9899	Text identification number range reserved for users. You must assign unique numbers.
	"text"		Text that appears in dialog

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, these two parameters must always be set to 0.

The following control characters may be inserted in text:

%n	Line break
%@x	Axis name of axis x (x is the axis number); HMI Embedded sl only Show axis names (HMI Embedded sl and HMI Advanced): NC access to corresponding machine data element containing the relevant axis name; text assembly by means of the string functions it contains.

Examples:

85000 0 0	"Retraction plane"
85001 0 0	"Drilling depth"
85002 0 0	"Pitch"
85003 0 0	"Pocket radius"

6.3 Storage structure of configuration files

6.3.1 HMI Embedded sl

Storage

In the Linux environment, the user configurations are copied to the **/user/sinumerik/hmi/proj** directory (for "normal" user dialogs, i.e., everything except cycles support) on the CF card. User configurations for user cycles support are copied to the **/user/sinumerik/cycles/proj** directory on the CF card. All files are **unpacked** and copied to the relevant directory.

Similarly, the manufacturer configurations are copied to the **/oem/**sinumerik/.... directories.

Path	Content
/card/oem/sinumerik/hmi/proj	User configurations (com files for "normal" user dialogs, i.e., everything except cycles support)
/card/user/sinumerik/cycles/proj	User configurations for user cycles support
/card/oem/sinumerik/cycles/proj /card/oem/sinumerik/hmi/proj	Manufacturer configurations
/card/user/sinumerik/cycles/ico/icoxxx /card/oem/sinumerik/cycles/ico/icoxxx /card/user/sinumerik/hmi/ico/icoxxx /card/oem/sinumerik/hmi/ico/icoxxx	Bitmaps
/card/user/sinumerik/hmi/cfg /card/oem/sinumerik/hmi/cfg	Ini files
/card/user/sinumerik/hmi/ico/icoxxx /card/oem/sinumerik/hmi/ico/icoxxx	Header icons
/card/user/sinumerik/cycles/lng/xxx /card/oem/sinumerik/cycles/lng/xxx /card/user/sinumerik/hmi/lng/xxx	The relevant texts (aluc.txt or aluctx.s0x), where xxx stands for the language

On delivery, three USB drives and access to the CF card are set up in the PROGRAM operating area.

The following directory structure is set up on the USB memory:

\cycles	
\cycles\proj	(com files)
\cycles\prog	(customer cycles (.spf))
\cycles\lng	(Language directory – contains only subdirectories)
\cycles\lng\xxx	(Language directory, e.g., ger, eng, etc. – one directory for each language. The **aluc.txt** file for the relevant language is stored here.)

\cycles\ico	(Display image directory – contains only subdirectories for each resolution)
\cycles\ico\ico640	Directory for the display images of resolution 640*480 as **.bmp** or **.bin**
\cycles\ico\ico800	
\cycles\ico\ico1024	

For dialogs, which are assigned to other operating areas:

\hmi\proj

\hmi\lng\...

\hmi\ico\...

In the PROGRAM operating area, the entire \cycles or \hmi directory is copied from the USB memory to the /user/sinumerik directory on the CF card.

6.3.2 HMI Advanced

Overview

When using HMI Advanced, no control file entries are required.

The directories are searched for the configuration files in the specified order. If files of the same name are stored in different directories, then the first file detected according to the search strategy is interpreted.

Test on a PG/standard PC

To test the configured dialogs on a PG/standard PC, the following conditions apply:

- You have installed the PC version of the HMI Advanced software on your PC/PG.
- The directory structure is the same as on HMI Advanced.
- The error log is generated in: \DH\COM.DIR\ERROR.COM

Storing the alarm text files

Alarm text files are stored in the following directory: \DH\MB.DIR\

File name: ALUC_*xx*.COM

References: HMI Advanced Startup Manual

6.4 Search function on the HW platform common to both HMI systems

6.4.1 Search function principle

Overview

HMI Embedded WIN32 searches the configuration files for "Expand user interface" on the same paths as HMI Advanced. The starting point is the data management path.

The search sequences described below are relevant for ShopMill/ShopTurn on NCU if HMI Advanced and HMI Embedded sl are both operated in parallel on a single hardware platform. In this case, "Expand user interface" accesses the same configuration files in ShopMill/ShopTurn on NCU running on HMI Embedded sl as on HMI Advanced.

Supplementary Conditions

HMI Embedded WIN32 evaluates the "mmchome"property in section [DHSTART] in the DH.NI configuration file containing the root directory for the data management path. DH.INI must be stored in the BIN directory path on which MMC0 .EXE is started. The maximum length for the data management root directory path entry is 100 characters.

Startup

On startup, HMI Embedded WIN32 uses the entries in the registry to ascertain where HMI Advanced is installed. Within the directory indicated in the registry, a search is made for the DH.INI file in the **..\user, ..\oem, ..\add_on, ..\mmc2** subdirectories in the sequence specified. If DH.INI cannot be found there, a search is carried out for the file in the currently selected directory of HMI Embedded WIN32.

Example

If HMI Advanced was found on F:\HMI\HMI Advanced, then a search for the DH.INI is made in the following sequence:

- F:\HMI\HMI Advanced\user
- F:\HMI\HMI Advanced\oem
- F:\HMI\HMI Advanced\add_on
- F:\HMI\HMI Advanced\mmc2

Control variables for search sequences

Search sequence with HMI Advanced:

- CUS directory in the data management path specified in dh.ini
- CMA directory in the data management path specified in dh.ini
- CST directory in the data management path specified in dh.ini
- COM directory in the data management path specified in dh.ini

Directory on the RAMDISK:

Filename.bin

Filename.bmp .

Bitmap name without path

In HMI Advanced, files with a .bin extension are not used

Search sequence if no archive is specified and the file name does not contain a path:

Filename.bin in the CUS directory on the data management path

Filename.bmp in the CUS directory on the data management path

Filename.bin in the *Filename.bi_* archive in the CUS directory on the data management path

Filename.bmp in the *Filename.bi_* archive in the CUS directory on the data management path

Filename.bin in the **CUS\resolution** directory on the data management path

Filename.bmp in the **CUS\resolution** directory on the data management path

Filename.bin in the *Filename.bi_* archive in the **CUS\resolution** directory on the data management path

Filename.bmp in the *Filename.bm_* archive in the **CUS\resolution** directory on the data management path

Filename.bin in the CMA directory on the data management path

Filename.bmp in the CMA directory on the data management path

Filename.bin in the *Filename.bi_* archive in the CMA directory on the data management path

Filename.bmp in the *Filename.bm_* archive in the CMA directory on the data management path

Filename.bin in the **CUS\resolution** directory on the data management path

Filename.bmp in the **CMA\resolution** directory on the data management path

Filename.bin in the *Filename.bi_* archive in the **CMA\resolution** directory on the data management path

Filename.bmp in the *Filename.bm_* archive in the **CUS\resolution** directory on the data management path

Filename.bin in the CST directory on the data management path

Filename.bmp in the CST directory on the data management path

Filename.bin in the *Filename.bi_* archive in the CST directory on the data management path

Filename.bmp in the *Filename.bm_* archive in the CST directory on the data management path

Filename.bin in the **CST\resolution** directory on the data management path

Filename.bmp in the **CST\resolution** directory on the data management path

Filename.bin in the *Filename.bi_* archive in the **CST\resolution** directory on the data management path

Filename.bmp in the *Filename.bm_* archive in the **CST\resolution** directory on the data management path

Filename.bin in the current directory (Bin directory)

Filename.bmp in the current directory (Bin directory)

Filename.bin in the *Filename.bi_* archive in the current directory (Bin directory)

Filename.bmp in the *Filename.bm_* archive in the current directory (Bin directory)

Filename.bin in the **CUS.ARJ** archive in the **CUS** directory on the data management path

Filename.bmp in the **CUS.ARJ** archive in the **CUS** directory on the data management path

Filename.bin in the **CUS.ARJ** archive in the **CUS\resolution** directory on the data management path

Filename.bmp in the **CUS.ARJ** archive in the **CUS\resolution** directory on the data management path

Filename.bin in the **CUS.ARJ** archive in the **CMA** directory on the data management path

Filename.bmp in the **CUS.ARJ** archive in the **CMA** directory on the data management path

Filename.bin in the **CUS.ARJ** archive in the **CMA\resolution** directory on the data management path

Filename.bmp in the **CUS.ARJ** archive in the **CMA\resolution** directory on the data management path

Filename.bin in the **CUS.ARJ** archive in the **CST** directory on the data management path

Filename.bmp in the **CUS.ARJ** archive in the **CST** directory on the data management path

Filename.bin in the **CUS.ARJ** archive in the **CST\resolution** directory on the data management path

Filename.bmp in the **CUS.ARJ** archive in the **CST\resolution** directory on the data management path

Steps 29 through 40 are then repeated for CMA.ARJ

Filename.bin in archive CMA.ARJ in the CUS directory on the data management path

....

Filename.bmp in the CMA.ARJ archive in the CST\resolution directory on the data management path

Steps 41 through 52 are then repeated for CST.ARJ

Filename.bin in archive CST ARJ in the CUS directory on the data management path

....

Filename.bmp in archive CST.ARJ in the CST\resolution directory on the data management path

Filename.bin in archive CUS.ARJ in the current directory (Bin directory)

Filename.bmp in archive CUS.ARJ in the current directory (Bin directory)

Filename.bin in archive CMA.ARJ in the current directory (Bin directory)

Filename.bmp in archive CMA.ARJ in the current directory (Bin directory)

Filename.bin in archive CST.ARJ in the current directory (Bin directory)

Filename.bmp in archive CST.ARJ in the current directory (Bin directory)

Filename.bxx

Partial binary files, which have to be processed for reasons of compatibility with Real mode and when starting Protected mode. These files are only searched for in the currently selected directory.

6.4.2 Search function for COMMON.COM

Storage location of COMMON.COM

The central control file COMMON.COM must be stored in one of the following NC directories:

- CUS: User cycles directory
- CMA: Manufacturer cycles directory
- CST: Standard cycles directory
- COM: Comment directory

The system searches through the directories for COMMON.COM in the following order. The first file with this name is evaluated.

In HMI Advanced, the COMMON.COM control file can also be stored on the data management path:

..\dh\cus.dir

..\dh\cma.dir

..\dh\cst.dir

..\dh\com.dir

Note

In order to activate the changes in this file, HMI Embedded WIN32 must be **restarted** in conjunction with HMI Advanced (ShopMill/ShopTurn).

6.4.3 Search function for images

Search sequence

The expanded search function for images (bitmaps) in HMI Embedded sl can also be used for images in the "Expand user interface" system.

If HMI Embedded WIN32 (ShopMill/ShopTurn) is installed together with HMI Advanced on a single hardware platform, a relevant data management path can be preset.

If a data management path is defined, it is prioritized and processed before the currently selected directory in order that HMI Embedded WIN32 and HMI Advanced access the same images.

The new search mechanism also includes the directories of the data management path that are of relevance for "Expand user interface", and the possible image archives (cus.arj, cma.arj, cst.arj).

The search mechanism always searches for the individual files first, before searching possible archives. Consequently, the search sequence for images is as follows:

- Individual image before archive (.bin files are searched for before .bmp files). Once the search of the individual files has been completed, the archives containing a single file (.bi_, then .bm_) are searched.
- Bitmap name with path followed by bitmap name without path

See also

Search function principle (Page 6-10)

Appendix A

A.1 List of start softkeys

Start softkeys for ShopMill and ShopTurn

ShopMill	SCxxx	Output dialog	
	SC818	Machine Manual operating area (large dialog)	Horizontal SK 8
	SC8181	Machine Manual operating area (medium dialog)	Horizontal SK 8
	SC8182	Machine Manual operating area (small dialog)	Horizontal SK 8
	SC8131	Machine Manual-Workpiece Zero operating area	Vertical SK 1
	SC8132	Machine Manual-Workpiece Zero operating area	Vertical SK 2
	SC8133	Machine Manual-Workpiece Zero operating area	Vertical SK 3
	SC8134	Machine Manual-Workpiece Zero operating area	Vertical SK 4
	SC8135	Machine Manual-Workpiece Zero operating area	Vertical SK 5
	SC8136	Machine Manual-Workpiece Zero operating area	Vertical SK 6
	SC8137	Machine Manual-Workpiece Zero operating area	Vertical SK 7
	SC8141	Machine Manual-Measure Tool operating area	Vertical SK 1
	SC8142	Machine Manual-Measure Tool operating area	Vertical SK 2
	SC8143	Machine Manual-Measure Tool operating area	Vertical SK 3
	SC8144	Machine Manual-Measure Tool operating area	Vertical SK 4
	SC8145	Machine Manual-Measure Tool operating area	Vertical SK 5
	SC8146	Machine Manual-Measure Tool operating area	Vertical SK 6
	SC8147	Machine Manual-Measure Tool operating area	Vertical SK 7

ShopMill	SCxxx	Output dialog	
	SC826	Machine Auto operating area (large dialog)	Horizontal SK 6
	SC8261	Machine Auto operating area (medium dialog)	Horizontal SK 6
	SC8262	Machine Auto operating area (small dialog)	Horizontal SK 6
	SC8426	Program-Drilling operating area	Vertical SK 6
	SC8436	Program-Milling operating area	Vertical SK 6
	SC8454	Program-Miscellaneous operating area	Vertical SK 4
	SC8951	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 1
	SC8952	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 2
	SC8953	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 3
	SC8954	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 4
	SC8955	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 5
	SC8956	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 6
	SC8957	Program-Miscellaneous-Workpiece Zero operating area	Vertical SK 7
	SC8961	Program-Miscellaneous-Measure Tool operating area	Vertical SK 1
	SC8962	Program-Miscellaneous-Measure Tool operating area	Vertical SK 2
	SC8963	Program-Miscellaneous-Measure Tool operating area	Vertical SK 3
	SC8964	Program-Miscellaneous-Measure Tool operating area	Vertical SK 4
	SC8965	Program-Miscellaneous-Measure Tool operating area	Vertical SK 5
	SC8966	Program-Miscellaneous-Measure Tool operating area	Vertical SK 6
	SC8967	Program-Miscellaneous-Measure Tool operating area	Vertical SK 7
	SC857	Messages/Alarms operating area	Horizontal SK 7
	SC858	Messages/Alarms operating area	Horizontal SK 8
	SC867	Tools/Zero offsets operating area	Horizontal SK 7
	SC8492	Program-G-Code-Editor operating area	Horizontal SK 2 (contour cycles support) *
	SC8493	Program-G-Code-Editor operating area	Horizontal SK 3 (drilling cycles support) *
	SC8494	Program-G-Code-Editor operating area	Horizontal SK 4 (milling cycles support) *

ShopMill	SCxxx	Output dialog	
	SC8495	Program-G-Code-Editor operating area	Horizontal SK 5 (turning cycles support) *
	SC8496	Program-G-Code-Editor operating area	Horizontal SK 6*
	SC8406	Program-G-Code-Editor operating area (expanded area)	Horizontal SK 6 (measuring cycles support)
	SC8407	Program-G-Code-Editor operating area (expanded area)	Horizontal SK 7 (measuring cycles support)

* These are Siemens dialogs.

ShopTurn	SCxxx	Output dialog	
	SC818	Machine Manual operating area (large dialog)	Horizontal SK 8
	SC8181	Machine Manual operating area (medium dialog)	Horizontal SK 8
	SC8182	Machine Manual operating area (small dialog)	Horizontal SK 8
	SC8131	Machine Manual-Workpiece Zero operating area	Vertical SK 1
	SC8132	Machine Manual-Workpiece Zero operating area	Vertical SK 2
	SC8133	Machine Manual-Workpiece Zero operating area	Vertical SK 3
	SC8134	Machine Manual-Workpiece Zero operating area	Vertical SK 4
	SC8135	Machine Manual-Workpiece Zero operating area	Vertical SK 5
	SC8136	Machine Manual-Workpiece Zero operating area	Vertical SK 6
	SC8137	Machine Manual-Workpiece Zero operating area	Vertical SK 7
	SC8141	Machine Manual-Measure Tool operating area	Vertical SK 1
	SC8142	Machine Manual-Measure Tool operating area	Vertical SK 2
	SC8143	Machine Manual-Measure Tool operating area	Vertical SK 3
	SC8144	Machine Manual-Measure Tool operating area	Vertical SK 4
	SC8145	Machine Manual-Measure Tool operating area	Vertical SK 5
	SC8146	Machine Manual-Measure Tool operating area	Vertical SK 6
	SC8147	Machine Manual-Measure Tool operating area	Vertical SK 7
	SC826	Machine Auto operating area (large dialog)	Horizontal SK 6

ShopTurn	SCxxx	Output dialog	
	SC8261	Machine Auto operating area (medium dialog)	Horizontal SK 6
	SC8262	Machine Auto operating area (small dialog)	Horizontal SK 6
	SC8246	Program-Drilling operating area	Vertical SK 6
	SC9436	Program-Turning operating area	Vertical SK 6
	SC9456	Program-Milling operating area	Vertical SK 6
	SC8454	Program-Miscellaneous operating area	Vertical SK 4
	SC857	Messages/Alarms operating area	Horizontal SK 7
	SC858	Messages/Alarms operating area	Horizontal SK 8
	SC867	Tools/Zero offsets operating area	Horizontal SK 7
	SC8492	Program-G-Code-Editor operating area	Horizontal SK 2 (contour cycles support) *
	SC8493	Program-G-Code-Editor operating area	Horizontal SK 3 (drilling cycles support) *
	SC8494	Program-G-Code-Editor operating area	Horizontal SK 4 (milling cycles support) *
	SC8495	Program-G-Code-Editor operating area	Horizontal SK 5 (turning cycles support) *
	SC8496	Program-G-Code-Editor operating area	Horizontal SK 6*
	SC8406	Program-G-Code-Editor operating area (expanded area)	Horizontal SK 6 (turning measuring cycles support)*
	SC8407	Program-G-Code-Editor operating area (expanded area)	Horizontal SK 7 (milling measuring cycles support)*

* These are Siemens dialogs.

A.2 List of colors

Available colors

A standard color table is available for HMI Advanced and HMI Embedded sl for the purpose of configuring dialogs (subset of the respective standard colors):

Number	Color
1	black
2	Red/brown
3	Dark green
4	Light gray
5	Dark gray
6	Blue
7	Red
8	brown
9	yellow
10	white

The appearance of the colors may vary slightly in each HMI program.

HMI Advanced

For bitmaps in HMI Advanced, the current color table of the shipped software must be used in the character program.

HMI Embedded sl

For bitmaps in HMI Embedded sl, the current color table of the shipped software must be used in the character program. Color tables depend on the "New Fashion" option.

You will find the color tables in the tool box under
8x0d\examples_tools\wizard.bsp\hmi_emb\...

The file names indicate the application of each table:

- HMI_EMB_NEW_FASHION.PAL:

  This color palette is used for HMI Embedded sl with New Fashion.

  Color indices 160 to 231 are available.

- HMI_EMB_OLD_AND_NEW_FASHION.PAL:

  This color palette is used for HMI Embedded sl with both Old and New Fashion; the bitmaps generated with this color palette are identical in appearance for Old and New Fashion. Color indices 160, 163, 184, 187, 196, 199, 204, 205, 207, 217, 219, 220, 221, 223, 226 and 228 are available

The previous color table HMI_EMB.PAL is superseded by the tables specified above. Only colors 160 through 231 can be used. This is the only way to ensure that images look the same on HMI Embedded sl and HMI Advanced.

Activating a color table in Paint Shop Pro:

- File → Open → ...*.bmp
- Colors → Open picture palette → ...*.PAL
- Apply palette by selecting the "Color Indices: Open" option.

System colors

There is a choice of 10 colors available for the color of an element (text, input field, background, etc.). The system colors have been expanded and there is a difference between the old and new designs (e. g., the header color).

In order to achieve a demarcation between unique and **design-dependent** colors, it has been defined that the unique colors are found between 0 and 128. The design-dependent colors that have been newly introduced as part of this expansion, are defined from index 128. This means, that also when the unique colors are expanded (up to 128), it is prevented that both color types are mixed.

Newly defined colors

Index	Color description	Color	
		Old design	New design
128	System color active field	yellow	orange
129	Background color	Gray	Light gray
130	Header color (active)	yellow	Blue
131	Header font color (active)	black	white

A.3 List of accessible system variables

Name	Index	Description
$A_DBB[x]	x=ByteNo	Data byte from/to PLC
$A_DBD[x]	x=Offset	Double data word (32bit) from/to PLC
$A_DBR[x]	x=Offset	Real data (32 bit) from/to PLC
$A_DBW[x]	x=Offset	Data word (16 bit) from/to PLC
$A_DLB[index]	index=Offset	Data byte in link area
$A_DLD[index]	index=Offset	Data access to double word in link area
$A_DLR[index]	index=Offset	Real data in link area
$A_DLW[index]	index=Offset	Data word in link area
$A_IN[x]	x=DigitalinputNo	Value of HW digital input
$A_INA[x]	x=AnaloginputNo	Value of HW analog input
$A_INCO[x]	x=InputNo	Comparator input NC
$A_INSE		Safe Programmable Logic: External input of the NCK I/O
$A_INSED		Safe Programmable Logic: Image of external NCK input
$A_INSEP		Safe Programmable Logic: External input of the PLC I/O
$A_INSEPD		Safe Programmable Logic: Image of external PLC inputs
$A_INSI		Safe Programmable Logic: Internal NCK input safety
$A_INSID		Safe Programmable Logic: Image of internal NCK input safety
$A_INSIP		Safe Programmable Logic: Internal PLC input safety
$A_LINK_TRANS_RATE		Link transfer rate
$A_MARKERSI		Safe Programmable Logic: NCK markers
$A_MARKERSIP		Safe Programmable Logic: Image of PLC markers
$A_OUT[x]	x=DigitaloutputNo	Value of HW digital output
$A_OUTA[x]	x=AnalogoutputNo	Value of HW analog output
$A_OUTSE		Safe Programmable Logic: External output of the NCK I/O
$A_OUTSED		Safe Programmable Logic: Image of external NCK output
$A_OUTSEP		Safe Programmable Logic: External output of the PLC I/O
$A_OUTSEPD		Safe Programmable Logic: Image of the external PLC output

Name	Index	Description
$A_OUTSI		Safe Programmable Logic: Internal NCK output safety
$A_OUTSID		Safe Programmable Logic: Image of internal NCK output safety
$A_OUTSIP		Safe Programmable Logic: Internal PLC output 611D safety
$A_OUTSIPD		Safe Programmable Logic: Image of internal PLC output 611D safety
$A_TIMERSI		Safe Programmable Logic: NCK timers
$A_PBB_IN[index]	index=Offset	IN data byte
$A_PBB_OUT[index]	index=Offset	OUT data byte
$A_PBD_IN[index]	index=Offset	IN double data word
$A_PBD_OUT[index]	index=Offset	OUT double data word
$A_PBR_IN[index]	index=Offset	IN real data
$A_PBR_OUT[index]	index=Offset	OUT real data
$A_PBW_IN[index]	index=Offset	IN data word
$A_PBW_OUT[index]	index=Offset	OUT data word
$A_TC_FCT		Command number
$A_TC_LFN		Source location number
$A_TC_LFO		Source location number
$A_TC_LTN		Target location number
$A_TC_LTO		Target location number
$A_TC_MFN		Source magazine
$A_TC_MFO		Source magazine number
$A_TC_MTN		Target magazine number
$A_TC_MTO		Target magazine number
$A_TC_STATUS		Command status
$A_TC_THNO		Number of toolholder
$A_TC_TNO		T number
$A_TOOLMLN[x]	x=ToolNo T	Current location
$A_TOOLMN[x]	x=ToolNo T	Current magazine
$AA_COUP_ACT[x]	x=Spindle following	Current coupling status following spindle
$AA_COUP_OFFS[x]	x=Axis	Offset to leading axis/leading spindle, setpoint
$AA_COUP_OFFS[x]	x=Spindle	Position offset for synchronous spindle (setpoint)
$AA_CURR[x]	x=Axis	Actual current value of axis or spindle
$AA_DELT[x]	x=Axis	Drive-specific distance to go in WCS
$AA_DTBB[x]	x=Axis	Drive-specific path from start of block in basic coordinate system
$AA_DTBW[x]	x=Axis	Drive-specific path from start of block in WCS
$AA_DTEB[x]	x=Axis	Drive-specific path, end of block in basic coordinate system

Name	Index	Description
$AA_DTEPB[x]	x=Axis	Drive-specific dist.-to-go infeed oscillation in basic coordinate system
$AA_DTEPW[x]	x=Axis	Drive-specific dist.-to-go infeed oscillation in WCS
$AA_DTEW[x]	x=Axis	Drive-specific path, end of block in WCS

$AA_EG_ACTIVE [a,b]	a = Following axis
	b = Leading axis
		EG coupling active

$AA_EG_AX[n,a]	n = Index leading axis
	a = Following axis
		EG leading axis number

$AA_EG_DENOM [a,b]	a = Following axis
	b = Leading axis
		EG coupling factor denominator

$AA_EG_NUM_LA[a]	a=Following axis	EG no. of leading axes

$AA_EG_NUMERA [a,b]	a = Following axis
	b = Leading axis
		EG coupling factor numerator

$AA_EG_SYN[a,b]	a = Following axis
	b = Leading axis
		EG synchronous position leading axis

$AA_EG_SYNCDIFF[a]	a=Axis identifier	EG synchronism deviation
$AA_EG_SYNFA[a]	a=Following axis	EG synchronous position following axis

$AA_EG_TYPE[a,b]	a = Following axis
	b = Leading axis
		EG coupling type

$AA_ESR_ENABLE[a]	a=Axis	ESR axis enable
$AA_ESR_ENABLE[a]	a=Axis	ESR enable
$AA_ESR_STAT[a]	a=Axis	ESR status
$AA_ETRANS[x]	x=FrameNo	Offset of external frames
$AA_FXS[x]	x=Axis	Status after travel to fixed stop
$AA_IBN[x]	x=Axis	Toolholder actual value
$AA_IEN[x]	x=Axis	Holder for active tool relative to workpiece zero point
$AA_IM[x]	x=Axis	Toolholder
$AA_IW[x]	x=Axis	Toolholder position, setpoint
$AA_LEAD_P[x]	x=Axis	Real master value - position
$AA_LEAD_SP[x]	x=Axis	Simulated master value - position
$AA_LEAD_SV[x]	x=Axis	Simulated master value - velocity
$AA_LEAD_TYP[x]	x=Axis	Source of master value
$AA_LEAD_V[x]	x=Axis	Real master value - velocity
$AA_LOAD[x]	x=Axis	Drive load in % (for 611D only)
$AA_MM[x]	x=Axis	Measured value in MCS
$AA_MM1[x]	x=Axis	Access to measured value in MCS
$AA_MM2[x]	x=Axis	Access to measured value in MCS
$AA_MM3[x]	x=Axis	Access to measured value in MCS
$AA_MM4[x]	x=Axis	Access to measured value in MCS
$AA_MW[x]	x=Axis	Measured value in WCS
$AA_OFF[x]	x=Axis	Overlaid motion for programmed axis
$AA_OFF_LIMIT[x]	x=Axis	Drive-specific limit value reached, correction for $AA_OFF

Name	Index	Description
$AA_OSCILL_REVERSE_POS1[x]	x=Axis	Current reversal position 1 oscill. in synchronous actions
$AA_OSCILL_REVERSE_POS2[x]	x=Axis	Current reversal position 2 oscill. in synchronous actions
$AA_OVR[x]	x=Axis	Drive-specific override for motion-synchronous actions
$AA_POWER[x]	x=Axis	Drive active power in [Watts]
$AA_S[x]	x=SpindleNo	Spindle speed, setpoint
$AA_SOFTENDN[x]	x=Axis	Software limit position, negative direction
$AA_SOFTENDP[x]	x=Axis	Software limit position, positive direction
$AA_STAT[x]	x=Axis	Axis status
$AA_SYNA_MEM		Free memory space, motion synchronous actions
$AA_SYNC[x]	x=Axis	Coupling of the slave axis for master value coupling
$AA_TORQUE[x]	x=Axis	Drive torque setpoint in [Nm]
$AA_TYP[x]	x=Axis	Axis type
$AA_VACTB[x]	x=Axis	Drive-specific feed, actual value
$AA_VACTM[x]	x=Axis	Drive-specific feed, setpoint
$AA_VACTW[x]	x=Axis	Drive-specific feed, actual value
$AA_VC[x]	x=Axis	Drive-specific feed, additive path feed correction
$AC_ALARM_STAT		ESR alarm status
$AC_AXCTSWA[CTn]	CTn=Axis container no.	Axis container status
$AC_DELT		Distance to go, path WCS
$AC_DRF[x]	x=Axis	DRF value
$AC_DTBB		Distance from start of block in basic coordinate system
$AC_DTBW		Distance from block start in WCS
$AC_DTEB		Distance from end of block in basic coordinate system
$AC_DTEPB		Distance to go for oscillating infeed of basic coordinate system
$AC_DTEPW		Distance to go for oscillating infeed in WCS
$AC_DTEW		Distance from block end in WCS
$AC_FCT0[x]	x=PolynomNo	a0 coefficient nth polynomial for synchr. action
$AC_FCT1[x]	x=PolynomNo	a1 coefficient nth polynomial for synchr. action
$AC_FCT2[x]	x=PolynomNo	a2 coefficient nth polynomial for synchr. action
$AC_FCT3[x]	x=PolynomNo	a3 coefficient nth polynomial for synchr. action
$AC_FCTLL[x]	x=PolynomNo	Lower limit value nth polynomial for synchronized action
$AC_FCTUL[x]	x=PolynomNo	Upper limit value nth polynomial for synchronized action

$AC_FIFOx[y]	x=FIFONo (1-10)
	y=ParameterNo
		FIFO variables for synchronous actions

Name	Index	Description
$AC_MARKER[x]	x=MarkerNo	Counter for motion synchronous actions
$AC_MEA		Probe has responded
$AC_OVR		Path override for synchronous actions
$AC_PARAM[x]	x=ParameterNo	Dyn. parameter for motion synchronization
$AC_PATHN		Normalized path parameter
$AC_PLTBB		Distance from start of block in basic coordinate system
$AC_PLTEB		Distance from block end in basic coordinate system
$AC_PRESET[x]	x=Axis	PRESET value
$AC_PROG		Program status
$AC_RETPOINT[x]	x=Axis	Return position on contour for repositioning
$AC_SDIR[x]	x=SpindleNo	Rotat. status
$AC_SMODE[x]	x=SpindleNo	Spindle mode
$AC_STAT		Channel status
$AC_TIME		Time from the start of the block in seconds (including the times for the internally generated intermediate blocks)
$AC_TIMES		Time from the start of the block in seconds (without times for the internally generated intermediate blocks)
$AC_TIMEC		Time from the start of the block in interpolation clock cycles (including clock cycles for the internally generated intermediate blocks)
$AC_TIMESC		Time from the start of the block in interpolation clock cycles (without the clock cycles for internally generated intermediate blocks)
$AC_TIMER[x]	x=TimerNo	Timer location in seconds
$AC_VACTB		Interpolation feedrate, setpoint
$AC_VACTW		Path velocity in WCS
$AC_VC		Additive path feed correction for synchronous actions
$AN_AXCTAS[n]		Actual axis container address (rotary position)
$AN_AXCTSWA[CTn]	CTn=Axis container no.	Axis container rotation active
$AN_ESR_TRIGGER		ESR start signal
$AN_MARKER[x]	x=MarkerNo	Marker variable for motion synchronous actions
$MC_DISPLAY_AXIS	Bits 16 -31	Machine axis identifier
$MC_DISPLAY_AXIS	Bits 0 -15	Identifier for geometry/auxiliary axis
$MC_MM_NUM_BASE_FRAMES		Number of base frames in channel
$MN_EXTERN_LANGUAGE $MN_EXTERN_CNC_SYSTEM	and	CNC system language
$MN_MAX_CUTTING_EDGE_NO		Max. value D number
$MN_MAX_CUTTING_EDGE_ PER_TOOL		Max. number of edges per tool

Name	Index	Description
$MN_MAX_SUMCORR_ PERCUTTING_EDGE		Max. number of total offsets per cutting edge
$MN_MM_KIND_OF_SUMCORR		Properties of total offsets in NCK
$MN_MM_NUM_CC_MAGAZINE_ PARAM		Number of parameters for one tool magazine
$MN_MM_NUM_CC_MAGLOC_ PARAM		Number of parameters for one tool magazine location
$MN_MM_NUM_CC_MON_ PARAM		Number of parameters of monitoring user data of one tool edge
$MN_MM_NUM_GLOBAL_BASE_ FRAMES		Number of channel-dependent basic frames
$MN_MM_NUM_SUMCORR		Total number of total offsets in NCK
$MN_MM_NUM_TOOL_ADAPTER		Max. number of traversed tool adapter data blocks
$MN_MM_TOOL_MANAGEMENT_ MASK		Settings NCK tool management
$P_UBFR[x ,MI]	x=FrameNo	Mirroring of settable frame
$P_UBFR[x,RT]	x=FrameNo	Rotation of settable frame
$P_UBFR[x,SC]	x=FrameNo	Scaling factor of settable frame
$P_UBFR[x,SI]	x=FrameNo	Fine offset for frames
$P_UBFR[x,TR]	x=FrameNo	Translation of settable frame
$P_CHBFRMASK		Active channel-specific basic frames
$P_EG_BC[a]		EG block change criterion
$P_NCBFRMASK		Active channel-independent basic frames
$P_OFFN		Offset normal
$P_PFRAME[x,TR] $P_ACTFRAME $P_IFRAME	or or	Compilation of active frame
$P_TOOL		Number of active tool edge
$P_TOOLL[1]		Active tool length 1
$P_TOOLL[2]		Active tool length 2
$P_TOOLL[3]		Active tool length 3
$P_TOOLND[x]	x=ToolNo	Number of cutting edges
$P_TOOLNO		Number of active tool
$P_TOOLR		Active tool radius
$P_UIFR[x,y,MI]	x=FrameNo, y=Axis	Mirroring of settable frame
$P_UIFR[x,y,RT]	x=FrameNo, y=Axis	Rotation of settable frame
$P_UIFR[x,y,SC]	x=FrameNo, y=Axis	Scaling factor of settable frame
$P_UIFR[x,y,SI]	x=FrameNo, y=Axis	Fine offset for frames
$P_UIFR[x,y,TR]	x=FrameNo, y=Axis	Translation of settable frame
$P_UIFRNUM		Index of active set frame
$R[x]	x=ParameterNo	R parameters
$SC_PA_ACTIV_IMMED[x]	x=Number protection zone	Immediately active after referencing
$SC_PA_CENT_ABS[x,0]	x=Number protection zone	Abscissa of arc center of 1st contour element

Name	Index	Description
$SC_PA_CENT_ABS[x,1]	x=Number protection zone	Abscissa of arc center of 2nd contour element
$SC_PA_CENT_ABS[x,2]	x=Number protection zone	Abscissa of arc center of 3rd contour element
$SC_PA_CENT_ABS[x,3]	x=Number protection zone	Abscissa of arc center of 4th contour element
$SC_PA_CENT_ABS[x,4]	x=Number protection zone	Abscissa of arc center of 5th contour element
$SC_PA_CENT_ABS[x,5]	x=Number protection zone	Abscissa of arc center of 6th contour element
$SC_PA_CENT_ABS[x,6]	x = Number protection zone	Abscissa of arc center of 7th contour element
$SC_PA_CENT_ABS[x,7]	x=Number protection zone	Abscissa of arc center of 8th contour element
$SC_PA_CENT_ABS[x,8]	x=Number protection zone	Abscissa of arc center of 9th contour element
$SC_PA_CENT_ABS[x,9]	x=Number protection zone	Abscissa of arc center of 10th contour element
$SC_PA_CENT_ORD[x,0]	x=Number protection zone	Ordinate of arc center of 1st contour element
$SC_PA_CENT_ORD[x,1]	x=Number protection zone	Ordinate of arc center of 2nd contour element
$SC_PA_CENT_ORD[x,2]	x=Number protection zone	Ordinate of arc center of 3rd contour element
$SC_PA_CENT_ORD [x,3]	x=Number protection zone	Ordinate of arc center of 4th contour element
$SC_PA_CENT_ORD[x,4]	x=Number protection zone	Ordinate of arc center of 5th contour element
$SC_PA_CENT_ORD[x,5]	x=Number protection zone	Ordinate of arc center of 6th contour element
$SC_PA_CENT_ORD[x,6]	x=Number protection zone	Ordinate of arc center of 7th contour element
$SC_PA_CENT_ORD[x,7]	x=Number protection zone	Ordinate of arc center of 8th contour element
$SC_PA_CENT_ORD[x,8]	x=Number protection zone	Ordinate of arc center of 9th contour element
$SC_PA_CENT_ORD[x,9]	x=Number protection zone	Ordinate of arc center of 10th contour element
$SC_PA_CONT_ABS[x,0]	x=Number protection zone	Abscissa of end point of 1st contour element
$SC_PA_CONT_ABS[x,1]	x=Number protection zone	Abscissa of end point of 2nd contour element
$SC_PA_CONT_ABS[x,2]	x=Number protection zone	Abscissa of end point of 3rd contour element
$SC_PA_CONT_ABS[x,3]	x=Number protection zone	Abscissa of end point of 4th contour element
$SC_PA_CONT_ABS[x,4]	x=Number protection zone	Abscissa of end point of 5th contour element
$SC_PA_CONT_ABS[x,5]	x=Number protection zone	Abscissa of end point of 6th contour element
$SC_PA_CONT_ABS[x,6]	x=Number protection zone	Abscissa of end point of 7th contour element
$SC_PA_CONT_ABS[x,7]	x=Number protection zone	Abscissa of end point of 8th contour element
$SC_PA_CENT_ABS[x,8]	x=Number protection zone	Abscissa of end point of 9th contour element
$SC_PA_CONT_ABS[x,9]	x=Number protection zone	Abscissa of end point of 10th contour element
$SC_PA_CONT_NUM[x]	x=Number protection zone	Number of valid contour elements
$SC_PA_CONT_ORD[x,0]	x=Number protection zone	Ordinate of end point of 1st contour element
$SC_PA_CONT_ORD[x,1]	x=Number protection zone	Ordinate of end point of 2nd contour element
$SC_PA_CONT_ORD[x,2]	x=Number protection zone	Ordinate of end point of 3rd contour element
$SC_PA_CONT_ORD[x,3]	x=Number protection zone	Ordinate of end point of 4th contour element
$SC_PA_CONT_ORD[x,4]	x=Number protection zone	Ordinate of end point of 5th contour element
$SC_PA_CONT_ORD[x,5]	x=Number protection zone	Ordinate of end point of 6th contour element
$SC_PA_CONT_ORD[x,6]	x=Number protection zone	Ordinate of end point of 7th contour element
$SC_PA_CONT_ORD[x,7]	x=Number protection zone	Ordinate of end point of 8th contour element
$SC_PA_CONT_ORD[x,8]	x=Number protection zone	Ordinate of end point of 9th contour element
$SC_PA_CONT_ORD[x,9]	x=Number protection zone	Ordinate of end point of 10th contour element
$SC_PA_CONT_TYP[x,0]	x=Number protection zone	Contour type of 1st contour element
$SC_PA_CONT_TYP[x,1]	x=Number protection zone	Contour type of 2nd contour element

Name	Index	Description
$SC_PA_CONT_TYP[x,2]	x=Number protection zone	Contour type of 3rd contour element
$SC_PA_CONT_TYP[x,3]	x=Number protection zone	Contour type of 4th contour element
$SC_PA_CONT_TYP[x,4]	x=Number protection zone	Contour type of 5th contour element
$SC_PA_CONT_TYP[x,5]	x=Number protection zone	Contour type of 6th contour element
$SC_PA_CONT_TYP[x,6]	x=Number protection zone	Contour type of 7th contour element
$SC_PA_CONT_TYP[x,7]	x=Number protection zone	Contour type of 8th contour element
$SC_PA_CONT_TYP[x,8]	x=Number protection zone	Contour type of 9th contour element
$SC_PA_CONT_TYP[x,9]	x=Number protection zone	Contour type of 10th contour element
$SC_PA_LIM_3DIM[x]	x=Number protection zone	Limitation of protection zone, applicate
$SC_PA_MINUS_LIM[x]	x=Number protection zone	Lower limit of protection zone, applicate
$SC_PA_ORI[x]	x=Number protection zone	Plane assignment of protection zone
$SC_PA_PLUS_LIM[x]	x=Number protection zone	Upper limit of protection zone, applicate
$SC_PA_T_W[x]	x=Number protection zone	Workpiece or tool-related protection zone
$SN_PA_ACTIV_ IMMED[x]	x=Number protection zone	Immediately active after referencing
$SN_PA_CENT_ABS[x,0]	x=Number protection zone	Abscissa of arc center of 1st contour element
$SN_PA_CENT_ABS[x,1]	x=Number protection zone	Abscissa of arc center of 2nd contour element
$SN_PA_CENT_ABS [x,2]	x=Number protection zone	Abscissa of arc center of 3rd contour element
$SN_PA_CENT_ABS[x,3]	x=Number protection zone	Abscissa of arc center of 4th contour element
$SN_PA_CENT_ABS[x,4]	x=Number protection zone	Abscissa of arc center of 5th contour element
$SN_PA_CENT_ABS[x,5]	x=Number protection zone	Abscissa of arc center of 6th contour element
$SN_PA_CENT_ABS[x,6]	x=Number protection zone	Abscissa of arc center of 7th contour element
$SN_PA_CENT_ABS[x,7]	x=Number protection zone	Abscissa of arc center of 8th contour element
$SN_PA_CENT_ABS[x,8]	x=Number protection zone	Abscissa of arc center of 9th contour element
$SN_PA_CENT_ABS[x,9]	x=Number protection zone	Abscissa of arc center of 10th contour element
$SN_PA_CENT_ORD[x,0]	x=Number protection zone	Ordinate of arc center of 1st contour element
$SN_PA_CENT_ORD[x,1]	x=Number protection zone	Ordinate of arc center of 2nd contour element
$SN_PA_CENT_ORD [x,2]	x=Number protection zone	Ordinate of arc center of 3rd contour element
$SN_PA_CENT_ORD[x,3]	x=Number protection zone	Ordinate of arc center of 4th contour element
$SN_PA_CENT_ORD[x,4]	x=Number protection zone	Ordinate of arc center of 5th contour element
$SN_PA_CENT_ORD[x,5]	x=Number protection zone	Ordinate of arc center of 6th contour element
$SN_PA_CENT_ORD[x,6]	x=Number protection zone	Ordinate of arc center of 7th contour element
$SN_PA_CENT_ORD[x,7]	x=Number protection zone	Ordinate of arc center of 8th contour element
$SN_PA_CENT_ORD[x,8]	x=Number protection zone	Ordinate of arc center of 9th contour element
$SN_PA_CENT_ORD[x,9]	x=Number protection zone	Ordinate of arc center of 10th contour element
$SN_PA_CONT_ABS[x,0]	x=Number protection zone	Abscissa of end point of 1st contour element
$SN_PA_CONT_ABS[x,1]	x=Number protection zone	Abscissa of end point of 2nd contour element
$SN_PA_CONT_ABS[x,2]	x=Number protection zone	Abscissa of end point of 3rd contour element
$SN_PA_CONT_ABS [x,3]	x=Number protection zone	Abscissa of end point of 4th contour element
$SN_PA_CONT_ABS[x,4]	x=Number protection zone	Abscissa of end point of 5th contour element
$SN_PA_CONT_ABS[x,5]	x=Number protection zone	Abscissa of end point of 6th contour element
$SN_PA_CONT_ABS[x,6]	x=Number protection zone	Abscissa of end point of 7th contour element
$SN_PA_CONT_ABS[x,7]	x=Number protection zone	Abscissa of end point of 8th contour element

Name	Index	Description
$SN_PA_CONT_ABS[x,8]	x=Number protection zone	Abscissa of end point of 9th contour element
$SN_PA_CONT_ABS[x,9]	x=Number protection zone	Abscissa of end point of 10th contour element
$SN_PA_CONT_NUM[x]	x=Number protection zone	Number of valid contour elements
$SN_PA_CONT_ORD[x,0]	x=Number protection zone	Ordinate of end point of 1st contour element
$SN_PA_CONT_ORD[x,1]	x=Number protection zone	Ordinate of end point of 2nd contour element
$SN_PA_CONT_ORD[x,2]	x=Number protection zone	Ordinate of end point of 3rd contour element
$SN_PA_CONT_ORD[x,3]	x=Number protection zone	Ordinate of end point of 4th contour element
$SN_PA_CONT_ORD[x,4]	x=Number protection zone	Ordinate of end point of 5th contour element
$SN_PA_CONT_ORD[x,5]	x=Number protection zone	Ordinate of end point of 6th contour element
$SN_PA_CONT_ORD[x,6]	x=Number protection zone	Ordinate of end point of 7th contour element
$SN_PA_CONT_ORD[x,7]	x=Number protection zone	Ordinate of end point of 8th contour element
$SN_PA_CONT_ORD[x,8]	x=Number protection zone	Ordinate of end point of 9th contour element
$SN_PA_CONT_ORD[x,9]	x=Number protection zone	Ordinate of end point of 10th contour element
$SN_PA_CONT_TYP[x,0]	x=Number protection zone	Contour type of 1st contour element
$SN_PA_CONT_TYP[x,1]	x=Number protection zone	Contour type of 2nd contour element
$SN_PA_CONT_TYP[x,2]	x=Number protection zone	Contour type of 3rd contour element
$SN_PA_CONT_TYP[x,3]	x=Number protection zone	Contour type of 4th contour element
$SN_PA_CONT_TYP[x,4]	x=Number protection zone	Contour type of 5th contour element
$SN_PA_CONT_TYP[x,5]	x=Number protection zone	Contour type of 6th contour element
$SN_PA_CONT_TYP[x,6]	x=Number protection zone	Contour type of 7th contour element
$SN_PA_CONT_TYP[x,7]	x=Number protection zone	Contour type of 8th contour element
$SN_PA_CONT_TYP[x,8]	x=Number protection zone	Contour type of 9th contour element
$SN_PA_CONT_TYP[x,9]	x=Number protection zone	Contour type of 10th contour element
$SN_PA_LIM_3DIM[x]	x=Number protection zone	Limitation of protection zone, applicate
$SN_PA_MINUS_LIM[x]	x=Number protection zone	Lower limit of protection zone, applicate
$SN_PA_ORI[x]	x=Number protection zone	Plane assignment of protection zone
$SN_PA_PLUS_LIM[x]	x=Number protection zone	Upper limit of protection zone, applicate
$SN_PA_T_W[x]	x=Number protection zone	Workpiece or tool-related protection zone
$TC_ADPT ...		Adapter data
$TC_ADPTx $TC_ADPTT	x=1 to 3	Number of parameters per adapter
$TC_DPCE		Transformed edge correction value

$TC_DPCx[y,z]	x=ParamNo
	y=ToolNo, z=EdgeNo
		User-defined tool edge parameter

$TC_DPx[y,z]	x=ParamNo
	y=ToolNo, z=EdgeNo
		Edge correction value

$TC_DPx[y,z]	x=ParamNo
	y=ToolNo, z=EdgeNo
		Transformed edge correction value

$TC_ECP ...		Transformed location-dependent setup correction
$TC_MAMP3		Wear compound strategy
$TC_MAP1		Type of magazine
$TC_MAP2		Identifier of the magazine

Name	Index	Description
$TC_MAP3		Status of magazine
$TC_MAP4		Link 1 between magazine and next magazine
$TC_MAP5		Link 2 between magazine and preceding magazine
$TC_MAP6		Magazine dimension
$TC_MAP9		Number of active wear group

$TC_MAPCx[y]	x=ParameterNo
	y=MagazineNo
		Magazine user data for a tool magazine

$TC_MOP1(x,y) … $TC_MOP15(x,y)	x=ToolNo
	y=EdgeNo
		Monitoring data for each tool edge

$TC_MOPCx[y,z]	x=ParamNo,
	y=T number
	z=Edge
		Magazine user data for a tool edge

$TC_MPPCx[y,z]	x= ParamNo
	y= MagazineNo
	z= MagPlaceNo
		Magazine location user data for a tool magazine

$TC_MPPx	x=1,...7	No. of parameters per magazine location
$TC_SCP...		Transformed location-dependent wear correction
$TC_SCP...		Location-dependent wear correction
$TC_SCPx	x=13,...21,....71	Number of total offset parameters per total offset block
$TC_TP1		DUPLO number
$TC_TP10		Tool search type for replacement tools
$TC_TP11		Tool information for HMI
$TC_TP2		Tool designation
$TC_TP3		Size to left in half locations
$TC_TP4		Size to right in half locations
$TC_TP5		Size upwards in half locations
$TC_TP6		Size downwards in half locations
$TC_TP7		Magazine location type
$TC_TP8		Tool status
$TC_TP9		Tool monitoring method

$TC_TPCx[y]	x=ParameterNo
	y = ToolNo
		User-defined tool parameter

$TC_TPG1		Spindle number
$TC_TPG2		Chaining rule
$TC_TPG3		Minimum wheel diameter
$TC_TPG4		Minimum grinding wheel width
$TC_TPG5		Current width of grinding wheel
$TC_TPG6		Maximum speed of grinding wheel
$TC_TPG7		Maximum peripheral speed of grinding wheel (SUG)
$TC_TPG8		Angle of inclination of inclined grinding wheel

Name	Index	Description
$TC_TPG9		Compensation parameter for GWPS
$VA_COUP_OFFS[x]	x=Axis	Offset to leading axis / leading spindle, actual value
$VA_IS[x]	x=Axis	Safe actual position of axis
$VA_VACTM[x]	x=Axis	Actual-value, load-end axis velocity in MCS system

A.4 List of PI services

Programming

Syntax	**PI_SERVICE***(service, n parameters)*
Service	PI service identifier
n parameters	List of PI service parameters. Individual parameters are separated by commas.

Service	Explanation
Parameter	**Example**
_N_ASUP_	An interrupt number in the specified channel is assigned to a parts program stored in the NCK (identified by path name and program name). This PI service is identical to program instruction 'SETINT'.
	Par1 Interrupt number (0 – 8) *Par2* Priority (0 – 8) *Par3* Liftfast (0, 1) *Par4* Block synchronization (0, 1) *Par5* Path data with max. 32 positions
	Assignment of interrupt 5 to program MPF_DIR/TEST_MPF in the current channel. The interrupt has priority 3 and is executed without rapid lift on the *PI_SERVICE("_N_ASUP__",5, 3, 0, 0,"/_N_MPF_DIR/_N_TEST_MPF")*
_N_CANCEL	All alarms with the "Cancel Alarm" classification can be acknowledged with this command. Individual acknowledgment for specific alarms is not possible.

	All alarms with the "Cancel Alarm" classification are reset. *PI_SERVICE("_N_CANCEL“)*

Service Parameter	Explanation Example
_N_CRCEDN	Create tool edge by specifying the cutting edge number. If the T number of an existing tool is entered in parameter "T Number" of this service, then a tool edge for the existing tool is created (in this case, the parameter "D number", i.e., the number of the edge to be created, has a value range of 1-9). If a positive T number is specified as a parameter and the tool for the T number entered does not exist, then the PI service is aborted. If a value of 0 is entered as the T number (model of absolute D numbers), then the D number values can range from 1 – 31999. The new cutting edge is set up with the specified D number. If the specified cutting edge already exists, then the PI service is aborted in both cases.
	Par1 T number *Par2* D number T number == 0 ==> 1 – 3199 T number > 0 ==> 1 – 9
	A tool edge with number 3 is created for the tool with number 17 in the current TO area. *PI_SERVICE("_N_CRCEDN",17,3)*
_N_CREACE	Create a new tool edge for a specified tool. The next-higher available D number is automatically set up. The following blocks in the active file system are affected by this PI service: Tool offsets TO: The relevant tool edge (with ZERO content) is set up Monitoring data TS: (If available) the relevant tool edge (with ZERO content) is set up User edge data TUE: (If available) the relevant tool edge (with ZERO content) is set up (SW releases NCK < 10.x)
	Par1 Tool number 1 to 31999

	After the tool with number 55 has been set up in T area 1, a further 2 edges are created for it. The tool therefore now possesses a total of 3 edges.
	PI_SERVICE("_N_CREATO",55)
	PI_SERVICE("_N_CREACE",55)
	PI_SERVICE("_N_CREACE",55)

_N_CREATO	Create a tool with a specific T number. The following blocks in the active file system are affected by this PI service: Tool directory TV: The tool is entered as an existing tool. Tool offsets TO: The first tool edge D1 (with ZERO content) is set up. User edge data TUE: (If available) the first tool edge D1 (with ZERO content) is set up. User tool data TU: (If available) an empty data block is made available for the tool.
	Par1 Tool number 1 to 31999

	After the tool with number 55 has been set up in T area 1, a further 2 edges are created for it. The tool therefore now possesses a total of 3 edges.
	PI_SERVICE("_N_CREATO",55)
	PI_SERVICE("_N_CREACE",55)
	PI_SERVICE("_N_CREACE",55)

Service Parameter	Explanation Example
_N_DELECE	Delete a cutting edge: If the T number of an existing tool is entered in parameter "T Number" of this PI service, then the tool edge for the relevant tool is deleted (in this case, the parameter "D number, i.e., the number of edge to be deleted, has a value range of 1-9). If a positive T number is specified as a parameter and the tool for the T number entered does not exist, then the PI service is aborted. If a value of 0 is entered as the T number (model of absolute D numbers), then the D number values can range from 1-31999. If the specified cutting edge does not exist, then the PI service is aborted in both cases.
	Par1 T number of the tool for which the associated tool cutting edge is to be deleted. A setting of 0 indicates that there is no reference to a specific tool (absolute D number). *Par2* Cutting edge number of tool cutting edge that is to be deleted. Range of values: T number == 0 ⇒ 1 – 31999 T number > 0 ⇒ 1 – 9
	The tool edge with number 3 for the tool with number 17 in the current TO area is deleted: *PI_SERVICE("_N_DELECE",17,3)*
_N_DELETO	Deletes the tool with all tool edges in all data blocks in which it is stored. The tool is also deleted in the following blocks (if available): TO, TU, TUE, TV, TG (type 400), TD, TS.
	Par1 Tool number 1 to 31999
	Deletion of tool with T number 21 in the current T area. *PI_SERVICE("_N_DELETO",21)*
	Par1 Identifier for search mode 1: Block search without calculation. 2: Block search with calculation 3: Search including the main block analysis
	Start search with calculation in current channel. To start the PI service in a meaningful manner, the data structure for the block search (block SPRAF ; addressing on HMI Embedded sl with variable services via /Channel/Search/..) must be filled in beforehand. *PI_SERVICE("_N_FINDBL",2)*
_N_LOGIN_	A password, which sets the current access level, is sent to the NCK.
	Par1 Password (precisely 8 characters; if there are fewer than 8 characters, blanks must be added)
	Transfers a password to the NCK, thus setting another access level. *PI_SERVICE("_N_LOGIN_","TESTWORD")*
_N_LOGOUT	The current access level is reset.

	The current access level is reset. *PI_SERVICE("_N_LOGOUT")*

Service Parameter	Explanation Example
_N_SETUFR	The SYSTEM OR USER variables 'linShift', 'mirrorImgActive', 'rotation' and 'scaleFact' in channel-specific data block FU can be used to define up to 8 zero offsets per channel. PI service _N_SETUFR must be called in order to activate these user-defined zero offsets.

	Activation of a user frame. *PI_SERVICE("_N_SETUFR")*

List of abbreviations

B

B.1 Abbreviations

A	Output
ASCII	American Standard Code for Information Interchange American coding standard for the exchange of information
BAG	Mode group
OPI	Operator Panel Interface
CAD	Computer-Aided Design
CNC	Computerized Numerical Control Computerized numerical control
CR	Carriage Return
DAC	Digital-to-Analog Converter
DB	Data Block in the PLC
DBB	Data Block Byte in the PLC
DBW	Data Block Word in the PLC
DBX	Data block bit in the PLC
DDE	Dynamic Data Exchange: Dynamic Data Exchange
DIN	Deutsche Industrie Norm (German Industry Standard)
DIR	DIRectory: List
DPM	Dual-Port Memory
DOS	Disk Operating System
DRAM	Dynamic Random Access Memory
DRF	Differential Resolver Function: Differential revolver function (handwheel)
DRY	DRY run: Dry run feedrate
DW	Data word
E	Input
EG	Expansion unit
ESR	Extended Stop and Retract
FRAME	Data block (FRAME)
FIFO	First In - First Out: Method of storing and retrieving data in a memory.
BP	Basic program
GUD	Global User Data Global user data
HD	Hard Disk Hard disk
HMI	Human Machine Interface: Controller user interface
MSD	Main Spindle Drive

Hardware	Hardware
IBN	Start-up
ICA	Interpolatory Compensation Interpolatory compensation
INC	Increment: Increment
INI	INItializing data Initializing data
IPO	Interpolator
ISO	International Standard Organization
JOG	JOGging: Setup mode
K1 .. C4	Channel 1 to channel 4
LED	Light-Emitting Diode: Light-emitting diode
LF	Line Feed
K_v	Servo gain factor
LUD	Local User Data: Local user data
MB	Megabyte
MCP	Machine Control Panel Machine control panel (→ MCP)
MD	Machine data
MDI	Manual Data Input: Manual input
MCS	Machine coordinate system
MLFB	Machine-readable product designation
MPF	Main Program File: NC part program (main program)
MPI	Multi-Point Interface Multiport Interface
MCP	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
ZO	Zero point shift
OEM	Original Equipment Manufacturer
OP	Operator Panel: Operating equipment
PCU	Programmable Control Unit
PCMCIA	Personal Computer Memory Card International Association: Standards body for memory cards
PG	Programming device
PLC	Programmable Logic Control:
REF	REFerence point approach function
REPOS	REPOSition function
ROV	Rapid Override: Input correction
RPA	R-Parameter Active: Memory area on the 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
SKP	SKiP: Skip block
SPF	SubProgram File: Subprogram

SRAM	Static RAM (non-volatile)
GWPS	Grinding wheel surface speed
SW	Software
SYF	SYstem Files System files
TEA	Testing Data Active: Identifier for machine data
TO	Tool Offset Tool offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
UFR	User frame
FD	Feed Drive
WCS	Workpiece coordinate system
TO	Tool offset
TC	Tool change
ZOA	Zero Offset Active: Identifier (file type) for work offset data

Glossary

Action

Anything configured within a → Method: → Functions, → Calculating variables, → Changing properties, etc.

User variable

Variables defined by the user in the → Parts program or data block.

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).

Menu tree

A group of interlinked → Dialogs

Group

Reload unit for → Configuration file

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.

Definition lines

Program section in which → Variables and → Softkeys are defined.

Editor

ASCII Editor with which characters can be entered in a file and edited.

Properties

Characteristics of an object (e.g., of a → Variable)

Input/output field

Also I/O field: for inputting or outputting variable values.

Start softkey

→ Softkey with which the first newly created → Dialog is started.

Event

Any action, which initiates execution of a → Method: Input of characters, actuation of → Softkeys, etc.

Focus

Part of the screen that is highlighted to identify the current → Element, e.g., where the cursor is positioned.

Function

Sequence of operations programmed in → Methods as a function of → Parameters.

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.

HSx

Horizontal → Softkey x

Interpreter

The interpreter automatically converts the defined code from the → Configuration file into a → Dialog and controls its use.

Machine data

Settings of properties of the SINUMERIK system made by Siemens/machine manufacturer/end user and stored in the system. The following types of auxiliary function are available:

$MN_... General NC machine data

$MC_... Channel-specific machine data

$MA_... Axis-specific machine data

$MM_... Operator panel machine data

In addition to these there are → Setting data und drive machine data.

Method

Programmed sequence of operations executed when a corresponding → Event occurs.

HMI Advanced, HMI Embedded sl

User interface for PLC

NC

Numerical Control: System components of a SINUMERIK system, which controls the motional sequences of axes on the basis of → Parts programs.

NC code

Permissible language elements used to write SINUMERIK → Parts programs.

NC function

→ PI service

User comment

Comment created automatically during → NC code generation.

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 from → HMI Advanced/HMI Embedded sl.

PLC

Programmable Logic Control: Programmable controller, which performs mainly logic operations in the SINUMERIK system.

PLC keys

PLC keys are provided as hotkeys via the PLC interface of the HMI software. The functions triggered by them in the HMI can be configured.

They take the form of MCP keys or evaluations of PLC signal logic operations. 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

Configuration file

File, which contains definitions and instructions that determine the appearance of → Dialogs and their → Functions.

Registers

Memory for data exchange between→ Dialogs.

Recompile

→ NC code sections can be generated in a → Parts 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.

Setting data

Data, which communicates the properties of the machine tool to the NC control, as defined by the system software. In contrast to → Machine data, setting data can be altered immediately by the operator.

ShopMill

Optimized SINUMERIK application and user interface for 2½D milling operations.

ShopTurn

Optimized SINUMERIK application and user interface for turning operations.

Simulation

Simulation of a → Parts program run without movement of the actual machine axes.

Softkey

Key on front of operator panel, which activates a function displayed on the assigned area of the screen.

Softkey labels

Text/image on the screen, which is assigned to a → Softkey.

Softkey menu

All horizontal or all vertical → Softkeys

Column index

Column number of an array

Standard application

→ User interfaces that are integrated into standard software.

System variable

NC variable providing the NC states for parts program accesses and accesses from → HMI.

Selecting

A program formulated in the NC language, which specifies motion sequences for axes and various special actions.

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.

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.

VSx

Vertical → Softkey x

Line index

Row number of an array

Access level

Graduated system of authorization, which makes the accessibility and utilization of functions on the → User interface dependent on the authorization rights of the user.

Index

SINUMERIK

SINUMERIK
840D sl/840Di sl/810D
Online Help (HE1)

Commissioning Manual

Valid for: control system
SINUMERIK 840D sl/840DE sl
SINUMERIK 840Di sl/840DiE sl
SINUMERIK 840D/840DE
SINUMERIK 840Di/840DiE
SINUMERIK 810D/810DE

Software	Software Version
HMI Advanced	7.3

11/2006
6FC5397-0DP10-0BA0

Safety Guidelines

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

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

Warning

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the 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
Automation and Drives
Postfach 48 48
90437 NÜRNBERG
GERMANY

Order No.: 6FC5397-0DP10-0BA0
Ⓟ 12/2006

Table of contents

Tables

Programming help systems 1

1.1 Overview

Programming help systems

The extendible programming help systems listed below are available on the PCU 50.3 controller:

- Quick "ASCII" help
- Extended "PDF" help
  - Additional help (PDF file) from the quick "ASCII" help
  - Additional help on input fields of cycle parameterization screens
  - ISO mode help

The functionality and extendibility of these help systems is described in the sections below.

Software

- HMI Advanced
- Adobe Reader 4.x (available free over the Internet/from specialist dealer)

  The Adobe Acrobat Reader 4.x software is supplied on the SINUMERIK 840D sl controller as standard.
- Adobe Acrobat 4.x

  You will need the Adobe Acrobat Distiller 4.x software in order to convert documents to PDF and for editing documents once they are available in this format.

Storage locations of help files

Programming help: F:\DH\CST.DIR\HLP.DIR (may contain HLP, PDF and TXT files)

Alarm/MD help: F:\MMC2\HLP (may contain HLP, PDF and TXT files)

OPI help: F:\HMI_ADV\HLP (may contain HLP, PDF and TXT files)

Principle of PDF help

The PDF help depends on the following components being available in the help file storage location:

- Call file, e.g., PG_GR.TXT, with the information:

  <PDF file name> <ID> <search term> for each call
- Associated PDF file

Proceed as follows to call up the help:

- Start Adobe Acrobat Reader (supplied with the controller)
- Jump to page number (must be defined by index in document)
- Search for term (Adobe software search function)

Creation procedure

- Define jumps in the help document (e.g., using standard index function of DTP systems) and generate
- Store jump file as text file
- Assign ID numbers to the programming commands
- Generate PDF file

Result:

- A call file (text file) with the information:

  <PDF file name> <ID> <search term> for each call
- One or more associated PDF files

Note

In the event of multiple jumps for the same topic, each page must be entered on a separate line.

Example:

840D_PG.PDF 44 155

840D_PG.PDF 45 180

840D_PG.PDF 46 199

Hint:

In the case of a larger volume of indices, the call file can be created automatically using Microsoft Excel:

- Insert "index file" (*.TXT) in Excel
- Insert column
- Enter the name of the PDF file on the first line
- Select the menu items "Edit" – "Fill" – "Down"

- Copy the data and
- Save as "TXT" file

The syntax below is left over:

<PDF file name> <ID> <search term>

840D_PG.PDF 44 155

Testing the jumps in the document

Proceed as follows to test whether all the search strings can be found:

- Start Adobe Acrobat
- Select the menu item "Edit" –> "Find" (CTRL + F)
- Enter the search string.

Note

If the search strings cannot be found, possible causes include:

- Incorrect input
- The texts were not converted properly during PDF generation and cannot be found by the search engine for that reason.

  *Analysis:*Select a string you are looking for using Adobe Acrobat, copy it and paste it into a Word document.

  Remedy:
- Generate the PDF again or
- Generate the postscript file using Adobe printer driver Adobe PS 5.1.1 and then convert it using Adobe Distiller.

Search sequence

The help adheres to the search sequence below:

- F:\USER\HLP
- F:\ OEM\HLP
- F:\ ADD_ON\HLP
- F:\MMC2\HLP
- F:\HMI_ADV\HLP
- F:\ DH\CUS.DIR\HLP.DIR (user directory)
- F:\ DH\CMA.DIR\HLP.DIR (manufacturer directory)
- F:\ DH\CST.DIR\HLP.DIR (Siemens directory)
- F:\ DH\COM.DIR\HLP.DIR

1.2 Quick "ASCII" help

You can call up a number of help functions in the editor to provide programming support with the editing of part programs via the *"Information"* key:

- Quick "ASCII" help
  - **Descriptive text for instructions**, context-sensitive based on the cursor position in the part program (e.g., G9 "Exact stop – deceleration")
  - **Overview of topics** (e.g., "Preparatory functions", "Motion commands", "Path action", etc.) to which instructions are assigned
  - **Overview of instructions with descriptive texts**
  - **Search** in help window via topic or text
  - **Transfer** of selected instruction to the editor
- Extended "PDF" help
  - Call up **additional help** (PDF file) **from the quick "ASCII" help**, e.g., jump to Programming Guide Fundamentals.
  - **Call up additional help on input fields of cycle screens** (PDF-file)

1.2.1 Extending/replacing the help function

The machine manufacturer/end user can integrate additional M and H functions, cycles or macros in the help system.

Text files designed for this purpose which are accessed by the help function can be extended (see Section "Extension of text files").

The paths of the generated ASCII text files can be parameterized or added in initialization file "IF.INI" or under *"Settings"* (user file name).

1.2.2 Entries in the file "IF.INI"

Overview

The following sections are available in the "IF.INI" file for parameterization of the help function:

- TECHNOLOGY
- TEXTFILES
- CONTEXT
- TEXTSEARCH

TECHNOLOGY section

The relevant technology category of the connected machine can be entered in the [TECHNOLOGY] section so as to limit the display of instruction texts to relevant technology entries, i.e., only entries pertinent to the relevant technology are displayed.

The following technology codes can be specified:

"m" = Milling

"t" = Turning

"g" = Grinding

"s" = Nibbling

"p" = Punching

"a" = Display all technologies

TEXTFILES section

The paths (path\name) used for the Short Help text files on which the help function is based are parameterized in the [TEXTFILES] section.

The help function accesses the following text files:

- Siemens "DH\CST.DIR\IFS_GR.COM"
- Machine manufacturer "DH\CMA.DIR\IFM_GR.COM" (any file name)
- End user "DH\CUS.DIR\USER_GR.COM" (any file name)

  (entered under *"Settings"*).

Note

The path for the Siemens text file "DH\CST.DIR\IFS_GR.COM" **must** exist, otherwise an error message is output.

There are different parameterization options for the names of the text files when assigning the text path.

The following table lists the parameterization options and their interpretation by the help function.

Parameters for the text file	Interpretation
FileName.com	The non-language-specific text file is read out
FileName_XX.com (XX = language code, e.g., "GR" for German, for language codes see "Other languages" under "Alarm help")	The language-specific text file is read out

Example: ENDUSER=CUS.DIR\IFM_.COM

First of all, an attempt is made to open the non-language-specific text file "IFM.COM" in the path "C:\DH\CUS.DIR\" but, if the non-language-specific text file does not exist, the relevant language-specific text file (e.g., "IFM_GR.COM") is opened instead.

Extract from TXT file "IFM_GR.COM":

```
...
a2//G00/Linear interpolation with rapid traverse (m) , quick help
a2//G01/Linear interpolation with feed (m)
...
a/20/"PGA.PDF",2,"="/=/Assignment operator with extended help
...
a/3/"PG.PDF",17,"ANG"/ANG/Contour angle (s) with extended cycle help
...
m/14/"PGZ.PDF",302,"CYCLE77"/CYCLE77/Circular spigot milling cycle
with extended cycle help
...
```

CONTEXT section

Increased context sensitivity of the help function can be selected in the [CONTEXT] section. This setting is activated with "1" and deactivated with "0" (entered under "Settings").

If increased context sensitivity is activated and the cursor is positioned to the left of, next to or on an instruction text, all instructions with the same initial letters are displayed when help is called.

For example, if the programmer has selected instruction "G1", all instructions beginning with "G1" (e.g., "G1", "G17", "G18", "G19") are displayed.

If increased context sensitivity has been deactivated with "0", then only the instruction actually selected (if available) is displayed.

TEXTSEARCH section

The text search type can be specified in the [TEXTSEARCH] section. The following search methods can be parameterized:

1 = Instruction texts only

2 = Descriptive texts only

3 = Instruction and descriptive texts

(entered under "Search").

File IF.INI

```
[TECHNOLOGY]
"m" = Milling
"t" = Turning
"g" = Grinding
"s" = Nibbling
"p" = Punching
"a" = Display all technologies
[TEXTFILES]
Siemens = DH\CST.DIR\IFS_GR.COM
Machine manufacturer = DH\CMA.DIR\IFM_GR.COM
Machine manufacturer = DH\CMA.DIR\IFI_GR.COM
End user = DH\CUS.DIR\USER_GR.COM
[CONTEXT]
= 0: Context sensitivity not active
= 1: Context sensitivity active
[TEXTSEARCH]
= 1: Text search for instruction texts only
= 2: Text search for descriptive texts only
= 3: Text search for instruction and descriptive texts
```

1.2.3 Extension of text files

Categories

These two categories are available in the text files:

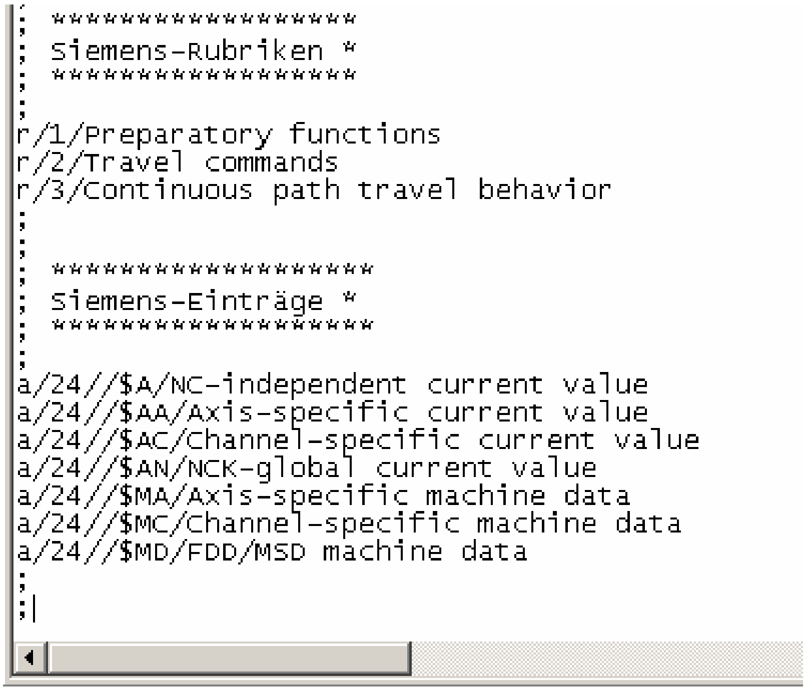
- Topic definition
- Instruction definition

The functions below can be entered in the *.COM text files of the help in the specified format:

- Topics as IDs of an **instruction grouping**
- Instructions with **descriptive texts**
- Instructions with call for extended help
  - **"PDF"** editor help or
  - Cycle help **on input fields of cycle screens**, in which parameters are supplied to the instruction
- Instructions with **alarm text variables** as descriptive texts

One line is available for each of these in the associated text file.

The screenshot below shows an example:



```
; ******************
; Siemens-Rubriken *
; ******************
;
r/1/Preparatory functions
r/2/Travel commands
r/3/Continuous path travel behavior
;
;
; *******************
; Siemens-Einträge *
; *******************
;
a/24//$A/NC-independent current value
a/24//$AA/Axis-specific current value
a/24//$AC/Channel-specific current value
a/24//$AN/NCK-global current value
a/24//$MA/Axis-specific machine data
a/24//$MC/Channel-specific machine data
a/24//$MD/FDD/MSD machine data
;
;
```

Figure 1-1 Siemens text file

The topic definition contains the following parameters:

Topic definition

`Technology code "r"/topic ID/topic name`

The instruction definition contains the following parameters:

Instruction definition

```
Technology/topic ID/extended help/instruction text/
descriptive text
```

Note

Within each of the individual text files, the Siemens topics are displayed first, followed by those of the machine manufacturer and finally those of the end user.

The specified sequence only appears if the **topics** are declared **first** and then the instructions of the text file.

The instructions in the text files must be stored in **alphabetical** order if the standard, machine manufacturer and end user instructions are to be sorted correctly.

1.2.4 Extension of a topic

Extension of a topic

A topic represents an instruction grouping which can be used for displaying instructions.

One line for the topic definition contains the following 3 parameters:

Topic definition

`Technology code "r"/topic ID/topic name`

The following line could then appear in the text file, for example:

`r/2/Motion commands`

`r:` Permanently defined in topic definitions

`2:` Topic ID used for assigning instructions.

`Motion commands:` Topic name

The "/" character separates the parameters and may only be used as a text character in the last parameter.

The length of the topic name is limited by the display and must not exceed 40 characters with a standard text size of 8 pt, for example.

The following topic IDs are reserved:

Table 1-1 Topic IDs

Topic ID	User
0..49	Siemens
50...99	Machine manufacturer
From 100	End user

The following topic IDs exist as standard in the Siemens text file:

Table 1-2 Defined topic IDs

Topic ID	Topic name	Topic ID	Topic name
1	Preparatory functions	13	Program handling
2	Motion commands	14	Standard cycles
3	Path action	15	Measuring cycles
4	Feedrate	16	Measurement
5	Spindle	17	Synchronized actions
6	Axis handling	18	Interrupt handling
7	M functions	19	String/field operations
8	H functions	20	Math. functions/operators
9	Offsets	21	Definitions
10	Transformations	22	High-level language elements
11	Tool management/ compensation	23	Parameters
12	Technology functions	24	System variables

1.2.5 Extension of an instruction

Extension of an instruction

An instruction contains descriptive texts, e.g., "G0/Linear interpolation with rapid traverse". An instruction is assigned to a topic via the topic ID.

Parameters of instruction definition

An instruction definition contains the following parameters:

Technology/topic ID/extended help/instruction text/descriptive text

Example of editor help:

```
a/20/"PGA.PDF",2,"="/=/Assignment operator
```

`a:` Milling *technology*

`20:` Standard cycles *topic ID*

`"PGA.PDF",2,"=":` *Extended help* on command "="

`=:` *Instruction text*

`Assignment operator:` *Descriptive text*

If the above line is entered in one of the user text files (e.g., IFM_GR.COM), help file "PGA.PDF" is called up for the command "=" and the relevant help section is displayed.

Example of cycle help:

```
m/14/"PGZ.PDF",302,"CYCLE83"/CYCLE83/Deep-hole drilling milling
cycle
```

`m:` Milling *technology*

`14:` Standard cycles *topic ID*

`"PGZ.PDF",302,"RTP":` *Extended help* for "RTP" parameter on the cycle parameterization screenform

`CYCLE83:` *Instruction text* CYCLE83

`Deep-hole drilling milling cycle:` *Descriptive text*

If the above line is entered in one of the user text files, user cycle "CYCLExx" is supported by the help system. It is displayed if

- either technology-independent code "a"
- or miller-specific code "m" has been entered in initialization file "IF.INI".

It is assigned to "Standard cycles" via topic ID "14" and can, therefore, also be found via the corresponding entry in the topic search.

The last parameter contains the descriptive text, which is displayed in the relevant column of the help system, and which is designed to give the user an initial indication as to the function of the cycle.

The "/" character separates the parameters and may only be used as a text character in the last parameter.

Technology

The following characters are defined for specifying the technology category:

"m" = Milling

"t" = Turning

"g" = Grinding

"s" = Nibbling

"p" = Punching

"a" = Display for all technologies

Note

Entries are separated by commas so as to enable them to be assigned to several technologies and different topics.

Topic ID

The topic ID is used for assigning instructions of a particular topic to the topic definition and for displaying these under the topic.

Integrating additional commands

If additional commands and a brief description are to be integrated in the help, these must be incorporated using the syntax described above in file IfM_gr.com (any file name), which is stored in directory Dh\CMA.dir\. If an additional help (extended "PDF" help) is to be called up for these commands, the parameter *"Extended help"* must be edited as described in the section below.

1.2.6 Parameterization of extended "PDF" help

Calling extended help

A special entry in the "Extended help" parameter of the instruction definition makes it possible to call up an extended "PDF" help as an online help in addition to the quick help. This may contain detailed information on the selected command, for example. The following calls are possible:

- Editor help:

  Jump directly to the help (PDF file) from the quick "ASCII" help for a term (command) in the program editor.

- Cycle help:

  Jump to the help (PDF file) from an input field of the cycle screen.

Parameters of instruction definition

Technology/topic ID/extended help/instruction text/descriptive text

Extended help

The following entries are possible in the "Extended help" parameter:

- Editor help

  ```
  /"PDF file name", help ID, "search string"/
  ```

- Cycle help

  ```
  /"PDF file name", help ID, "search string" (file name, help
  screen)/
  ```

Two commas must be inserted in the "Extended help" parameter even if no entries need to be made.

Example:

The declaration line requires the following parameters, for example:

```
m,t/14/"PDF filename", help ID, "search string" (file name, help
screen)/CYCLE82/drilling cycle, drilling, counterboring
```

The meanings of the entries for the "Extended help" parameter are as follows:

`PDF file name`

= File from which the descriptive text is to be displayed

`Help ID`

= Unique assignment to term for which the help is to be displayed

`Search string`

= Search string in the PDF file which is to be displayed in the help text

`(File name, help screen)`

= Name of file in which the help screens are stored and the name of the help screen which is to be displayed (-> only relevant for cycle help)

`CYCLE82`

= Instruction text

`Drilling cycle, drilling, counterboring`

= Descriptive text

Call example:

```
m,t/14/"PGZ.PDF",1,"RTP"(drilling,CYC82)/CYCLE82/drilling cycle,
drilling, counterboring
```

Note

If you only want to access the PDF file, only these parameters should then appear in the "Extended help" section.

E.g.: m,t/14/**"PGZ.PDF",1,"RTP"**/CYCLE82/drilling cycle, drilling, counterboring

If you only want to display the help screen, only these parameters should then appear in the "Extended help" section.

E.g.: m,t/14/,,**(drilling,CYC82)**/CYCLE82/drilling cycle, drilling, counterboring

Instruction text

An instruction text could be "G17", for example. This text is transferred to the part program currently being edited when the *"Transfer to editor"* command is issued.

The instruction text can be approx. 12 characters long.

Descriptive text

The descriptive text is the help text which describes the instruction in more detail.

The length of the descriptive text is limited by the display and must not exceed 40 characters with a standard text size of 8 pt, for example.

Multiple jumps

Several help references can be stored in ring formation (i.e., the first help is displayed again after the last help) for each help element.

E.g.: `m,t/14/"Hlp.pdf",1,"Help1",,2,"Help2","Help3.pdf",3,,/ CYCLE82/drilling cycle, drilling, counterboring`

I.e.:

- Entry in help text ring

  File: HLP.PDF

  Index: 1

  Text: Help1
- Entry in help text ring

  File: HLP.PDF

  Index: 2

  Text: Help2
- Entry in help text ring

  File: HELP3.PDF

  Index: 3

  Text: Help2

The components of a block are remembered by the previous block, as illustrated in the example, so as to avoid repetitions. Only complete blocks are transferred.

Note

Two commas must be inserted in the "Extended help" parameter even if no entries need to be made.

1.2.7 Extension with alarm text variables

Function

A special situation with respect to parameterization of the section for descriptive texts or topic names is to store these in the section instead of the descriptive texts ("alarm text variables").

These variables begin with "$".

E.g., m/14//CYCLExx/$888888

Instead of entering the descriptive text directly in the help text file, this is entered under alarm text variable "$888888", for example.

The actual declaration of the descriptive text is therefore stored in a user-specific alarm text file.

The result is a non-language-specific text file.

1.3 Extended "PDF" help

1.3.1 Function

You can call up a number of help functions in the editor to provide programming support with the editing of part programs via the *"Information"* key:

- Quick "ASCII" help
  - Descriptive text for instructions, context-sensitive based on the cursor position in the part program (e.g., G9 "Exact stop – deceleration")
  - Overview of topics (e.g., "Preparatory functions", "Motion commands", "Path action", etc.) to which instructions are assigned
  - Overview of instructions with descriptive texts
  - Search in help window via topic or text
  - Transfer of selected instruction to the editor
- Extended "PDF" help
  - Call up additional help (PDF file) from the quick "ASCII" help, e.g., jump to Programming Guide Fundamentals.
  - Call up additional help on input fields of cycle screens (PDF-file)

1.3.2 Extending programming help

A special entry in the "Extended help" parameter of the instruction definition makes it possible to integrate an extended help in addition to the quick help (e.g., detailed information on the selected command).

The following calls are possible:

- Editor help:

  Jump directly to the help (PDF file) from the quick "ASCII" help for a term (command) in the program editor.

  Preconditions:

  A list of additional commands which are to be displayed in the quick help has already been created as described in Section "Extension of text files" (e.g., IFM_GR.COM) and is stored in the controller. This allows you to call up the extended help after the quick help using the Info key. Please note that the list of commands for the quick help is identical to the list for the extended help (concordance).

- Cycle help:

  Jump to the help (PDF file) from an input field of the cycle screen.

Parameters of instruction definition

```
Technology/topic ID/extended help/instruction text/
descriptive text
```

```
"Extended help":
```

The following entries are possible in the "Extended help" parameter:

- Editor help

  `/"PDF file name", help ID, "search string"/`

  Example: `a/20/"PGA.PDF",2,"="/=/Assignment operator`

- Cycle help

  ```
  /"PDF file name", help ID, "search string" (file name, help
  screen)/
  ```

  Example: `a/3/"PG.PDF",17,"ANG"/ANG/Contour angle (s)`

Necessary files

- IFM_GR.COM: Text file with program commands and brief description
- *.PDF: Help file, e.g., 840D_PG_GR.PDF
- *.txt: Jump file, e.g., 840D_PG_GR.TXT

1) Define jump labels in document

The jumps must be identified by an index entry and a single-column index list (right-justified page numbers) must be generated in the document (e.g., Winword) from which sections are to be displayed on the controller as a PDF.

If several jumps are required for one command, the same number of index labels must be created.

2) Generate text file

Copy the single-column index generated above (right-justified page numbers) and save it in a text file (*.TXT).

Example:

MEAW 155

<Search string> <page>

The text file for the PDF call requires the following syntax:

<PDF file name> <ID> <page>

Enter the name of the PDF file **(without the language code)** at the start of each line.

Example:

840D_PG.PDF MEAW 155

<PDF name> <ID> <page>

3) Define ID numbers

ID numbers unique to each document (e.g., 1...n) must be defined for the search strings. The ID numbers correspond to the list of instructions of the quick "ASCII" help, as the detailed descriptions corresponding to the brief descriptions are to be displayed.

Example:

840D_PG.PDF 44 MEAW 155

<PDF file name> <ID> <search string> <page>

Now delete the "search string" column such that the syntax below remains:

Result:

<PDF file name> <ID> <page>

840D_PG.PDF 44 155

4) Generate PDF file

Print your document as a postscript file and convert this to PDF format using Adobe Acrobat Distiller.

5) Read files into controller

The generated files (*.PDF and *.TXT) must now be read into the controller, e.g., into directory F:\ DH\CUS.DIR\HLP.

Note

In the case of a larger volume of indices, the call file can be created automatically using Microsoft Excel:

- Insert text file (*.TXT) in Excel
- Insert column
- Enter the name of the PDF file on the first line
- Select "Edit > Fill"
- Now delete the "search string" column such that the syntax below remains:

  <PDF file name> <ID> <page>

  840D_PG.PDF 44 155
- Copy the data and
- Save as "TXT" file

1.4 ISO mode help

1.4.1 Function

Function

The 840D/840D sl-specific help systems are displayed as standard:

- Help in the editor
- Alarm help
- Machine data help

If ISO dialect 0 help systems are to be displayed, a distinction must be made as to which data is to be displayed, as the same commands can have multiple meanings, for example.

The [TEXTFILES] section of the "IF.INI" file has been extended by the entry SiemensI=CST.DIR\IFIT_COM (turning technology) or SiemensI=CST.DIR\IFIM_COM (milling technology) for this reason. This specifies the path (CST.DIR) and the name (IFIT_COM or IFIM_COM) for the ISO mode quick "ASCII" help.

The extended "PDF" help can thus be called from the quick "ASCII" help.

IF.INI [TEXTFILES]

```
[TEXTFILES]
;text files containing the information shown in the help system
Siemens=cst.dir\ifs_.com
SiemensI=cst.dir\ifit_.com or ifim_.com
Manufacturer=
End user=
```

Necessary files

- List of "ISO dialect 0" commands with the associated brief description
- TXT file for the jump to the PDF file
- PDF file which is to be displayed in the help

Program execution

The help checks the following machine data on startup:

- MD18800 MM_EXTERN_LANGUAGE

  If the MD is activated, the controller switches over to the ISO dialect 0 mode.

  If MD18800 = 1, MD20150 is evaluated.
- MD20150 GCODE_RESET_VALUES[n(active channel),47]

  If MD20150 [47] = 2, the file entered in the INI file under "SiemensI" (ISO mode) is used (otherwise the file under "Siemens" is used).
- MD10880 MM_EXTERN_CNC_LANGUAGE

  MD10880 specifies whether the controller is to be designed for turning or milling in ISO mode.

1.4.2 Extending ISO mode help

If additional descriptions are to be displayed in the ISO mode help, these must be inserted in the form specified below:

- As a list of ISO mode commands with a brief description in the directory

  "SiemensI=CST.DIR\" as file "IFIT_COM" or file "IFIM_COM"

  (refer to Section "Extension of text files")
- As a help description (PDF file) and a TXT file of the same name in directory "F:\USER\HLP"

Parameters of instruction definition

```
Technology/topic ID/extended help/instruction text/
descriptive text
```

```
"Extended help":
```

The following entries are possible in the "Extended help" parameter:

- Editor help

  ```
  /"PDF file name", help ID, "page number"/
  ```
- Cycle help

  ```
  /"PDF file name", help ID, "page number" (file name, help screen)/
  ```

Necessary files

- IFIT_.COM or IFIM_.COM: Text file with ISO mode commands and brief description
- *.PDF: e.g., 840D_PGIT_GR.PDF
- *.TXT: e.g., 840D_PITG_GR.TXT

1) Define jump labels in document

The jump labels must be identified by an index entry and a single-column index list (right-justified page numbers) must be generated in the document (e.g., Winword) which is to be displayed on the controller as a PDF.

2) Generate text file

Copy the single-column index generated above (right-justified page numbers) and save it in a TXT file.

Example:

G65 155

<Search string> <page>

The text file for the PDF call now requires the following syntax:

<PDF file name> <search string> <page>

Enter the name of the PDF file **(without the language code)** at the start of each line.

Example:

840D_PGIT.PDF G65 155

<PDF name> <search string> <page>

3) Define ID numbers

ID numbers unique to each document (e.g., 1...n) must be defined for the search strings. The ID numbers correspond to the list of instructions of the quick "ASCII" help, as the detailed descriptions corresponding to the brief descriptions are to be displayed.

Example:

840D_PGIT.PDF 44 MEAW 155

<PDF file name> <ID> <search string> <page>

Now delete the "search string" column such that the syntax below remains:

Result:

<PDF file name> <ID> <page>

840D_PGIT.PDF 44 155

4) Generate PDF file

Print your document as a postscript file and convert this to PDF format using Adobe Acrobat Distiller.

5) Read files into controller

The generated files (*.PDF and *.TXT) must now be read into directory F:\USER\HLP on the controller.

Note

In the case of a larger volume of indices, the call file can be created automatically using Microsoft Excel:

- Insert text file (*.TXT) in Excel
- Insert column
- Enter the name of the PDF file on the first line
- Select "Edit > Fill"
- Now delete the "search string" column such that the syntax below remains:

  <PDF file name> <ID> <page>

  840D_PGIT.PDF 44 155
- Copy the data and
- Save as "TXT" file

Alarm/Machine data help 2

2.1 Overview

The functionality and extendibility of the alarm and machine data help are described in the sections below.

Software

- HMI Advanced
- Adobe Reader 4.x (available free over the Internet/from specialist dealer)

  The Adobe Acrobat Reader 4.x software is supplied on the SINUMERIK 840D/840D sl controller as standard.
- Adobe Acrobat 4.x

  You will need the Adobe Acrobat Distiller 4.x software in order to convert documents to PDF and for editing documents once they are available in this format.

Storage locations of help files

Programming help: F:\DH\CST.DIR\HLP.DIR (may contain HLP, PDF and TXT files)

Alarm/MD help: F:\MMC2\HLP (may contain HLP, PDF and TXT files)

OPI help: F:\HMI_ADV\HLP (may contain HLP, PDF and TXT files)

Principle of PDF help

The PDF help depends on the following components being available in the help file storage location:

- Call file, e.g., PG_GR.TXT, with the information:

  <PDF file name> <ID> <search term> for each call
- Associated PDF file

Proceed as follows to call up the help:

- Start Adobe Acrobat Reader (supplied with the controller)
- Jump to page number (must be defined by index in document)
- Search for term (Adobe software search function)

Creation procedure

- Define jumps in the help document (e.g., using standard index function of DTP systems) and generate
- Store jump file as text file
- Generate PDF file

Result:

- A call file (text file) with the information:

  <PDF file name> <ID> <search term> for each call
- One or more associated PDF files

Note

In the event of multiple jumps for the same topic, each page must be entered on a separate line.

Example:

840D_PG.PDF 44 155

840D_PG.PDF 45 180

840D_PG.PDF 46 199

Hint:

In the case of a larger volume of indices, the call file can be created automatically using Microsoft Excel:

- Insert "index file" (*.TXT) in Excel
- Insert column
- Enter the name of the PDF file on the first line
- Select the menu items "Edit" – "Fill" –> "Down"
- Copy the data and
- Save as "TXT" file

The syntax below is left over:

<PDF file name> <ID> <search term>

840D_PG.PDF 44 155

Testing the jumps in the document

Proceed as follows to test whether all the search strings can be found:

- Start Adobe Acrobat
- Select the menu item "Edit" –> "Find" (CTRL + F)
- Enter the search string.

Note

If the search strings cannot be found, possible causes include:

- Incorrect input
- The texts were not converted properly during PDF generation and cannot be found by the search engine for that reason.

  *Analysis:*Select a string you are looking for using Adobe Acrobat, copy it and paste it into a Word document.

  Remedy:

  – Generate the PDF again or

  – Generate the postscript file using Adobe printer driver Adobe PS 5.1.1 and then convert it using the Adobe Distiller.

Search sequence

The help adheres to the search sequence below:

- F:\USER\HLP
- F:\ OEM\HLP
- F:\ ADD_ON\HLP
- F:\MMC2\HLP
- F:\HMI_ADV\HLP
- F:\ DH\CUS.DIR\HLP.DIR (user directory)
- F:\ DH\CMA.DIR\HLP.DIR (manufacturer directory)
- F:\ DH\CST.DIR\HLP.DIR (Siemens directory)
- F:\ DH\COM.DIR\HLP.DIR

2.2 Alarm help

2.2.1 Function

When help is called up, a PDF file is opened and the relevant alarm is displayed on the controller in Adobe Reader.

The files containing the alarms are stored in the directories below:

- Siemens alarms: F:\HMI_ADV\HLP
- Manufacturer alarms: F:\DH\CMA.DIR\HLP
- User alarms: F:\DH\USER\HLP

File names

The file names below must be adhered to (must **not** be changed), even for the purposes of additional alarms:

- ALARM_GR.PDF [German alarms]
- ALARM_UK.PDF [English alarms]
- ALARM_FR.PDF [French alarms]
- ALARM_IT.PDF [Italian alarms]
- ALARM_SP.PDF [Spanish alarms]

The names of the associated TXT files (contain the jump list) must therefore be as follows:

- ALARM_GR.TXT [German TXT file]
- ALARM_UK.TXT [English TXT file]
- ALARM_FR.TXT [French TXT file]
- ALARM_IT.TXT [Italian TXT file]
- ALARM_SP.TXT [Spanish TXT file]

Other languages

The alarm texts are available in the following languages:

CH: Simplified Chinese

TW: Standard Chinese

KO: Korean

JA: Japanese

SW: Swedish

HU: Hungarian

PO: Portuguese

CZ: Czech

TR: Turkish

RU: Russian

PL: Polish

NL: Dutch

The naming convention specified above for language codes must be used for the file names of the user alarms (must **not** be changed).

2.2.2 Extending alarm help

If additional user alarms are to be displayed in the alarm help, these must be inserted in directory C:\USER\HLP in the form of a PDF file and a TXT file of the same name.

Necessary files

- *.PDF: E.g.: ALARM_GR.PDF
- *.TXT: e.g., ALARM_GR.TXT

1) Define jump labels in document

The jump labels must be identified by an index entry and a single-column index list (right-justified page numbers) must be generated in the document (e.g., Winword) which is to be displayed on the controller as a PDF.

2) Generate text file

Copy the single-column index generated above (right-justified page numbers) and save it in a TXT file.

Example:

10702 90

<ID> <page>

The search string (alarm number) is identical to the ID number in this case.

The text file for the PDF call now requires the following syntax:

<PDF file name> <ID> <page>

Enter the name of the PDF file at the start of each line.

Example:

ALARM.PDF 10702 90

ALARM.PDF 10703 90

ALARM.PDF 10704 91

<PDF name> <ID> <page>

3) Generate PDF file

Print your document as a postscript file and convert this to PDF format using Adobe Acrobat Distiller.

4) Read files into controller

The generated files (*.PDF and *.TXT) must now be read into the controller, e.g., into directory F:\DH\USER\HLP.

2.2.3 User comments on help

Function

The alarm help ([i] key in diagnostics) provides the "Notepad" function for entering user-specific texts for individual alarms. These texts are stored in directory:

F:\USER\RH

Note

Alarm-specific texts can be stored if the HMI help is called up using the [i] key from the alarm overview or from the alarm log for the corresponding alarm.

This is not possible when the alarm is selected from the table of contents.

2.2.4 Changing "Win-Help" alarm help to "PDF help"

To change the help function from "Win-Help" to "PDF help", proceed as described in in the previous section: "Extension of alarm help".

1. **Define jump labels for each alarm number in the document:**

   The jump labels must be identified by an index entry and a single-column index list (right-justified page numbers) must be generated in the document (e.g., Winword) which is to be displayed on the controller as a PDF.

2. **Generate text file:**

   Copy the single-column index generated above (right-justified page numbers) and save it in a TXT file.

   The text file for the PDF call requires the following syntax:

   <PDF file name> <ID> <page>

   The search string (alarm number) is identical to the ID number in this case.

3. Enter the name of the PDF file at the start of each line.

   Example:

   ALARM.PDF 10702 90

   ALARM.PDF 10703 90

   ALARM.PDF 10704 91

   <PDF name> <ID> <page>

4. Print the **file** in which the alarm texts are stored **as a postscript file** (possible from all standard platforms). The file extension used in Winword is *.prn and the one used in Interleaf is *.ps, for example.

5. Convert the postscript file to "PDF" using the Adobe Acrobat Distiller software.

6. **Read** the **files** (*.PDF and *.TXT) into directory F:\DH\USER\HLP.DIR on the controller.

Note

If additional user alarms are to be displayed in the alarm help, these must be inserted in directory F:\USER\HLP in the form of a PDF file and a TXT file of the same name.

Please adhere to the naming conventions listed in the next section under "File names".

2.3 Machine data help

2.3.1 Function

The machine data help is based on the "PDF help", i.e., when help is called up, a PDF file is opened and the relevant alarm is displayed on the controller in Adobe Reader.

The files containing the machine data are stored in the directories below:

- Siemens MD: F:\MMC2\HLP
- User MD: F:\USER\HLP

File names

The file names below must be adhered to (must not be changed), even for the purposes of additional user MD:

- N_NCK_GR.PDF [German], N_NCK_UK.PDF [English]
- C_NCK_GR.PDF, C_NCK_UK.PDF
- A_NCK_GR.PDF, A_NCK_ UK.PDF
- S_NCK_GR.PDF, S_NCK_ UK.PDF
- DISPL_GR.PDF, DISPL_ UK.PDF
- DROT_GR.PDF, DROT_ UK.PDF
- DLIN_GR.PDF, DLIN_ UK.PDF
- HYDRM_GR.PDF, HYDRM_ UK.PDF

The names of the associated TXT files (contain the jump list) must therefore be as follows:

- N_NCK_GR.TXT [German], N_NCK_UK. TXT [English]
- C_NCK_GR. TXT, C_NCK_UK. TXT
- A_NCK_GR. TXT, A_NCK_ UK. TXT
- S_NCK_GR. TXT, S_NCK_ UK. TXT
- DISPL_GR. TXT, DISPL_ UK. TXT
- DROT_GR. TXT, DROT_ UK. TXT
- DLIN_GR. TXT, DLIN_ UK. TXT
- HYDRM_GR. TXT, HYDRM_ UK. TXT

Structure of text file

PDF file without language code	MD number	Page
DROT.PDF	1000	1
DROT.PDF	1001	1
DROT.PDF	1002	1
...	...	...

Meanings of names

Name	Meaning
N_NCK_GR.PDF	General MD
C_NCK_GR.PDF	Channel-specific MD
A_NCK_GR.PDF	Axis-specific MD
S_NCK_GR.PDF	Setting data
DISPL_GR.PDF	Display MD
DROT_GR.PDF	Drive MD, rotary
DLIN_GR.PDF	Drive MD, linear
HYDRM_GR.PDF	Drive MD, hydraulic module

2.3.2 Extending the machine data help

If additional user machine data are to be displayed in the MD help, these must be inserted in directory C:\USER\HLP in the form of a PDF file and a TXT file of the same name.

Necessary files

- *.PDF: e.g., N_NCK_GR.PDF
- *.TXT: e.g., N_NCK_GR.TXT

1) Define jump labels in document

The jump labels must be identified by an index entry and a single-column index list (right-justified page numbers) must be generated in the document (e.g., Winword) which is to be displayed on the controller as a PDF.

2) Generate text file

Copy the single-column index generated above (right-justified page numbers) and save it in a TXT file.

Example:

10000 170

<ID> <page>

The search string (MD number) is identical to the ID number in this case.

The text file for the PDF call now requires the following syntax:

<PDF file name> <search string> <page>

Enter the name of the PDF file at the start of each line.

Example:

N_NCK.PDF 10000 170

N_NCK.PDF 10002 170

N_NCK.PDF 10010 171

<PDF name> <ID> <page>

3) Generate PDF file

Print your document as a postscript file and convert this to PDF format using Adobe Acrobat Distiller.

4) Read files into controller

The generated files (*.PDF and *.TXT) must now be read into directory F:\USER\HLP on the controller.

Index

SIEMENS

SINUMERIK

SINUMERIK 840D sl/840Di sl/840D/840Di/810D Generating target language texts (TX2)

Commissioning Manual

Valid for:

Controls
SINUMERIK 840D sl/840DE sl
SINUMERIK 840D powerline/840DE powerline
SINUMERIK 840Di powerline/840DiE powerline
SINUMERIK 810D powerline/810DE powerline

Software	*Release:*
HMI Advanced	V7.3
HMI Embedded	V7.2

11/2006
6FC5397-0DP10-0BA0

Safety Guidelines

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

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

Warning

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the 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
Automation and Drives
Postfach 48 48
90437 NÜRNBERG
GERMANY

Order No.: 6FC5397-0DP10-0BA0
℗ 12/2006

Table of contents

Tables

Introduction 1

1.1 General

Since the 1990s, Microsoft have put considerable effort into satisfying global demands for information processing and software development.

In addition to supporting as many languages as possible, one of the aims was to be able to support all languages with the same uniform operating system, as opposed to the language-specific versions of operating systems which were used before.

As part of these efforts, many of the system components and applications were gradually converted to the cross-language platform Unicode.

Although in many cases the user would have been completely unaware of this conversion process, in terms of the subjects discussed in this guide fundamentally different procedures need to be adopted depending on which version of the operating system is used.

The procedure recommended for Windows 2000/XP is much simpler than the one recommended for Windows 95/NT4. Therefore it makes sense to change over to the new systems. The guide for older systems is no longer being updated by the documentation department.

1.2 Prerequisites

Supplementary conditions

This document describes the procedure for generating a text with the following preconditions:

- The text is to be generated on a Microsoft Windows operating system (Windows XP).
- The text is to be generated in a language not identical to the language of the operating system (= default language).
- The present description uses German as the default language for all examples provided herein. However, the described procedure will work just as well with a different language as the default Windows language.
- The text is to be generated in a format known as a "text file", i.e., a file which does not contain any formatting such as font type, font size, underlining, etc. This means that, apart from the text characters, the file will only contain ASCII control characters such as Tab, CR and LF (tab indent, carriage return and line feed).
- The language of the text requires a different Windows code page to that of the default language, i.e., the target language does not belong to the same language family as the default language.

- The text file is to be coded using this code page, not as a Unicode file.
- The text should be generated as simply as possible: in particular, this means that if possible, no additional software should have to be installed.
- For the purposes of this guide, it is assumed that you have installed Microsoft Word 2003 (or a later version) and are familiar with using it.

Text types for HMI Embedded and HMI Advanced

The following text types can be changed in the individual systems:

Text types	Systems
Alarm texts Texts for "Expand user interface" Application texts for all operating areas incl. softkey texts	HMI Embedded
Alarm texts Texts for "Expand user interface" Texts for the "Help" function in the editor Machine data Softkey texts for all operating areas	HMI Advanced

1.3 Language families

A language family is a group of languages for which Microsoft operating systems use the same code page. Microsoft uses the term "language group".

Table 1-1 Language families

Language family	Windows code page	Languages
Central European	1250	Albanian, Croat, Polish, Romanian, Serbian (Latin), Slovakian, Slovenian, Czech, Hungarian
Cyrillic	1251	Bulgarian, Macedonian, Russian, Serbian (Cyrillic), Ukrainian, White Russian
Western	1252	German, English, Finnish, French, Indonesian, Icelandic, Italian, Dutch, Norwegian, Portuguese, Swedish, Spanish
Greek	1253	Greek
Turkish	1254	Turkish
Baltic	1257	Estonian, Latvian, Lithuanian
Japanese	932	Japanese
Chinese	936	Simplified Chinese
Korean	949	Korean
Chinese	950	Standard Chinese (Traditional Chinese)

Operating system and target language from the same language family

If the language of the operating system (e.g., German) belongs to the same language family as the target language (e.g., Portuguese), then some difficulties could possibly occur with the **input** of certain characters. However, there will be no problems **displaying** these characters with a simple text editor.

Remedy:

- Use the Windows "Character Map" tool

  (see Section "Input via the Character Map") or
- or switch the keyboard

  (see Section "Using a different keyboard assignment") or
- use the ALT+digit combination to enter these characterseinzelne Zeichen mit ALT+Ziffernkombination eingeben

  (see Section "Numerical text input")

There should not be any further problems in this case. Use a simple text editor (e.g., Notepad) and save the texts in the usual way.

Section "Files with Word 2003" does not apply.

Operating system and target language from different language families

If the language of the operating system (e.g., German) does not belong to the same language family as the target language (e.g., Hungarian), difficulties will occur with both **inputting** and **displaying** the characters in a simple text editor (Notepad).

1.4 Tools

Standard version

This document describes the use of Microsoft Word 2003 and other tools which may not necessarily be installed on your PC.

You should have installed the following components:

- Microsoft Word 2003 (required)
- Windows XP: Language Support for East Asian characters (necessary for these languages)
- Windows XP: Character Map (recommended)

Checking the installation of the Language Support for East Asian languages

To install the Language Support for East Asian languages, please refer to Section "Special considerations when working with East Asian languages".

1.5 Terms

Code page

A list of characters which forms the character set for one or more languages. A code which is unique within this code page is assigned to each of the characters in the list.

All of the code pages used in Windows share the ASCII range (codes 0 to 127).

A distinction is made between single-byte code pages and multiple-byte code pages.

In single-byte code pages (all European languages), each character is represented with a single byte. Accordingly, single-byte code pages comprise a maximum of 256 characters.

Multiple-byte code pages (Asian languages) contain both characters which are stored as a single byte and characters which are stored with two (or more) bytes. The shared ASCII range is contained as a range of single-byte characters.

There is a 1:1 assignment between Windows code page and language family.

The language families are listed in Section "Language families".

Diacritical characters

(diacritical = distinguishing)

In most cases, a relatively small supplementary character, which is attached to a letter to give the letter a specific stress, a specific pronunciation or even a new meaning.

Section "Special characters in different language families" contains a list of diacritical characters.

If the combination of basic letters and diacritical characters belongs in a language-specific alphabet, this combination is contained in the appropriate Windows character sets (code page, Unicode) as a separate character.

Use of diacritical characters

A diacritical character is used

- To designate the combined character (e.g., Ģ = G cedilla, Õ = O tilde),
- For the combined input, if there is no separate key that possesses the required combined character

  (see Section "Using a different keyboard assignment"),
- In Vietnamese and Thai, for normal coding in text files,
- For output (only in typographically compound texts): Diacritical characters and basic letter are output separately, meaning that any combinations are possible, e.g., for ancient languages and phonetics.

In all the other cases, the combined letter is always considered as a compound unit.

In some cases, however, the combined letter is incorrectly referred to as the diacritical character.

Input Method Editor (IME)

An aid for inputting the CJK characters (CJK: Abbreviation for Chinese, Japanese and Korean).

Language family

A language family is a group of languages which uses the same code page. This requires the use of the same basic alphabet (e.g., Latin or Cyrillic), but it does not mean that the alphabet of all languages of this group is the same.

There is a 1:1 assignment between Windows code page and language family.

The language families are listed in Section "Language families".

Font files (TrueType fonts)

Windows XP does not use font files that are stored separated for each code page; instead it uses something known as "Big Fonts", which are fonts that contain the character set for several code pages. saving memory capacity for the common characters.

With Word 2003, the selection of characters from this total character set is performed via the Unicode coding, not via the code page coding. This makes it impossible to mix up characters with the same code in different code pages.

Default language

The term "default language" is used in this guide for the language of the operating system, i.e., German for a German Windows environment and English for an English Windows environment.

Default keyboard setting

Under "Control Panel" → "Regional and Language Options", in the tab "Languages", click under the heading "Text Services and Input languages" on the button "Details..." to reach the dialog box "Text Services and Input languages".

Here you can see the default keyboard setting in the "Settings" tab under the heading "Default Language and Region Support".

The default keyboard setting consists of a language property (left) and a keyboard assignment (right).

This language property of the default keyboard setting need not absolutely match the default language.

Text file

The term "Text file" is used to describe a file which does not contain any formatting (font type, font size, underscore, bold, italics etc.) or embedded objects (pictures, tables, graphics, footnotes etc.).

This means that, apart from the text characters, the file will only contain ASCII control characters like Tab, CR and LF (tab indent, carriage return and line feed).

The text characters are taken from a particular code page.

Occasionally, text files are also incorrectly referred to as ASCII files. This is only correct if a text file only contains ASCII characters (codes 0 to 127) and is therefore code page invariant.

Unicode

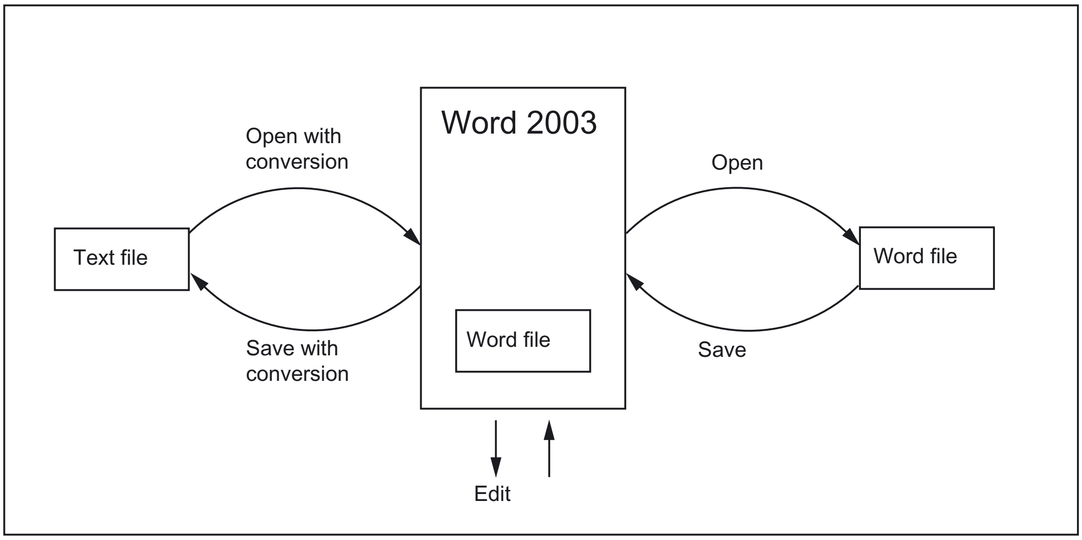
A 16-bit character set for which code pages are no longer required. All character codes are unique, without needing to specify the code page. Unicode programs like Word 2003 do not offer code page-dependent interpreting of characters and thus also cannot display characters differently by assigning a font or language property.

Editing files with Word 2

2.1 Overview

This section describes how text files can be edited using Microsoft Word 2003.

Word 2003 does not edit text files directly. Instead, it converts them to Word files upon opening. During saving Word can then convert the edited text back into a text file.	We recommend saving an additional version of the text file as a Word file and using this version as the basis for future editing. It is then no longer necessary to open and convert the file.



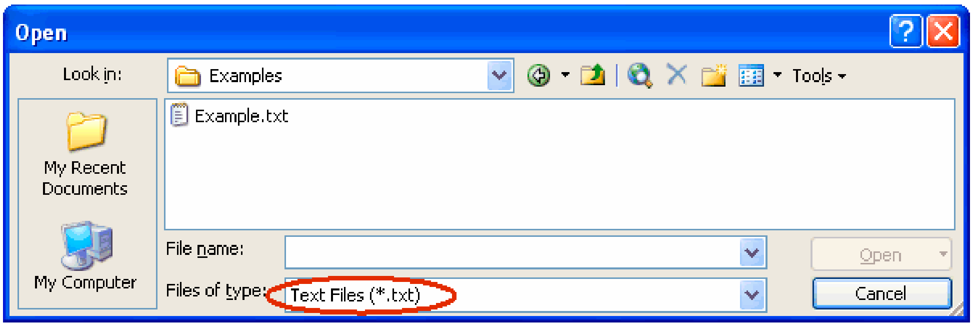
2.2 Generating a new text file

1. You can create a new text file as follows:
   - In Windows Explorer, click the command "File" → "New" → "Text Document".
   - Launch Notepad without a file. A new empty file is then opened. Then select the command "File" → "Save As" → "Encoding: ANSI"
   - Make a copy of an existing text file you wish to use as a template for the new text file.
2. Now open the text file with Word 2003 in the way described in the next section.

2.3 Opening an existing file

1. In Word 2003, select "Open…" from the "File" menu.
2. From the "Open" dialog box, choose the "Text files" (*.txt)" setting under "Files of type".

   If your file has a different extension from "txt", you must rename it.



File name:

Files of type: Text Files (*.txt)

3. Choose the file you would like to open and click "Open".

   Instead of steps 1-3, you can also use one of the standard shortcuts:

   – In Word you can use the list of the recently used files.

   – Drag and drop the text file from Explorer to the Word icon on your desktop.

   – If Word has already been launched, drag and drop the text file from Explorer to the title bar of the Word window.

4. Note what happens next.

   Continue with "Convert File" (see point 5.) if this dialog box opens.

   Continue with "File Conversion" (see point 6.) if this dialog box opens.

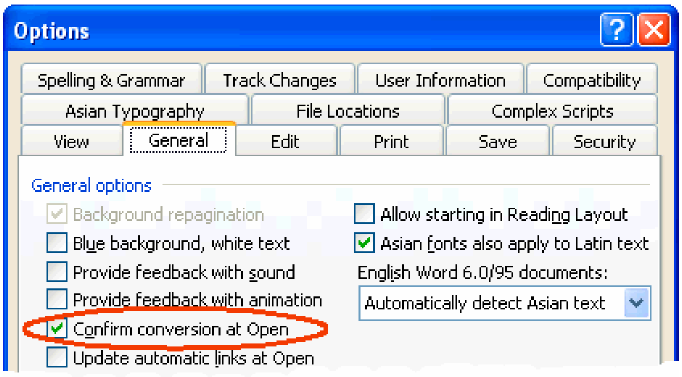
   If, however, the file is opened straightaway with neither the "Convert File" nor the "File Conversion" dialog boxes opening up, then Word has probably opened the file incorrectly under the assumption that the text has been written using the default language.

   Be careful with special characters. The special characters will appear incorrectly if the code page of the text file is not the same as the code page of the default language.

   In this case, go to "Extras" → "Options..." and open the Options dialog box for Word. Click "General" and select the option "Confirm conversion at Open".

   When this option is active the additional dialog box "Convert File" (see point 5.) will appear every time, but when the option is inactive it is skipped.

   Afterwards close the file and start again from point 1.



Options

Spelling & Grammar | Track Changes | User Information | Compatibility
Asian Typography | File Locations | Complex Scripts
View | General | Edit | Print | Save | Security

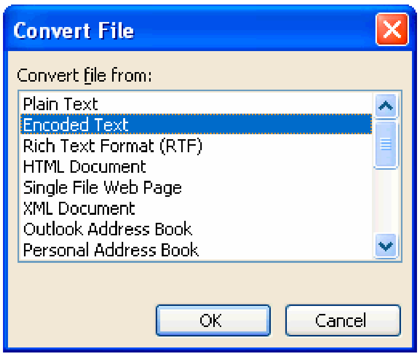
General options

- ☑ Background repagination
- □ Blue background, white text
- □ Provide feedback with sound
- □ Provide feedback with animation
- ☑ Confirm conversion at Open
- □ Update automatic links at Open
- □ Allow starting in Reading Layout
- ☑ Asian fonts also apply to Latin text

English Word 6.0/95 documents: Automatically detect Asian text

5. The "Convert File" dialog box will appear.

   Select "Encoded Text" and click "OK".



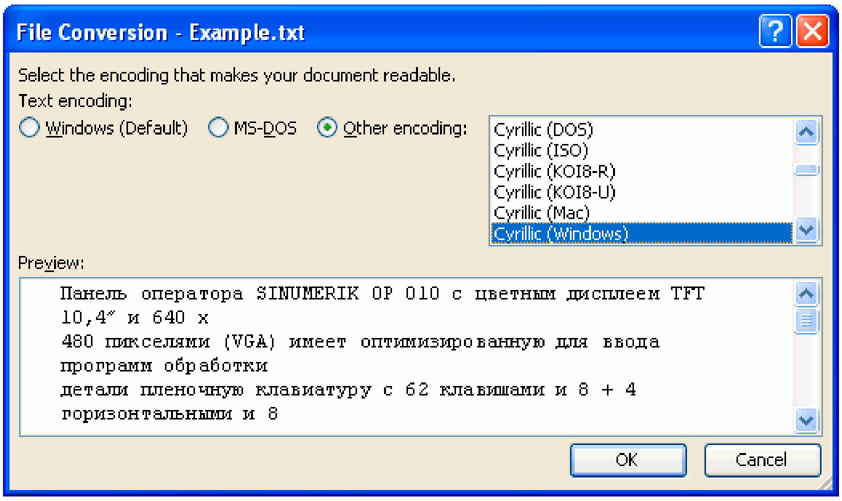
Convert File

Convert file from:

- Plain Text
- Encoded Text
- Rich Text Format (RTF)
- HTML Document
- Single File Web Page
- XML Document
- Outlook Address Book
- Personal Address Book

6. The "File Conversion - Example.txt" dialog box will then appear:

   – If "Windows (default)" is selected, activate "Other Encoding". Select the correct encoding (Word will make a suggestion, which will not necessarily be correct).



File Conversion - Example.txt

Select the encoding that makes your document readable.

Text encoding:

○ Windows (Default) ○ MS-DOS ● Other encoding:

- Cyrillic (DOS)
- Cyrillic (ISO)
- Cyrillic (KOI8-R)
- Cyrillic (KOI8-U)
- Cyrillic (Mac)
- Cyrillic (Windows)

Preview:

```
Панель оператора SINUMERIK OP 010 с цветным дисплеем TFT
10,4″ и 640 x
480 пикселями (VGA) имеет оптимизированную для ввода
программ обработки
детали пленочную клавиатуру с 62 клавишами и 8 + 4
горизонтальными и 8
```

   – Section "Codes for SINUMERIK HMI", Table 2-1 lists the suitable encoding options. Check the preview area while doing this.

7. Click "OK".

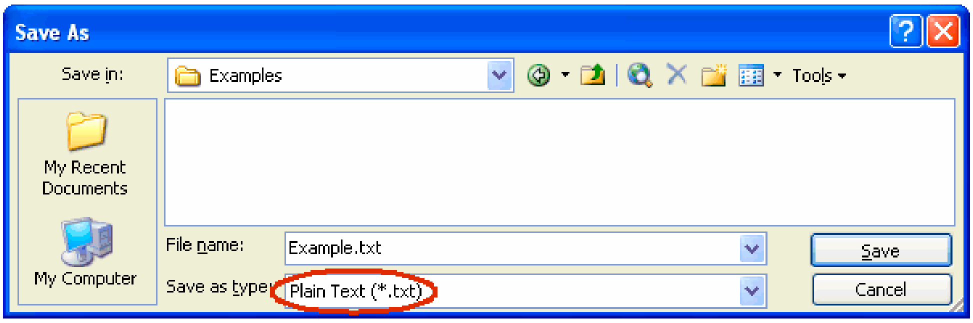
   The file is opened and converted to a Word file at the same time.

   The existing text is formatted using a default font.

2.4 Saving text files

Save the text entered as a "Text File":

1. In Word, select "Save As…" from the "File" menu.



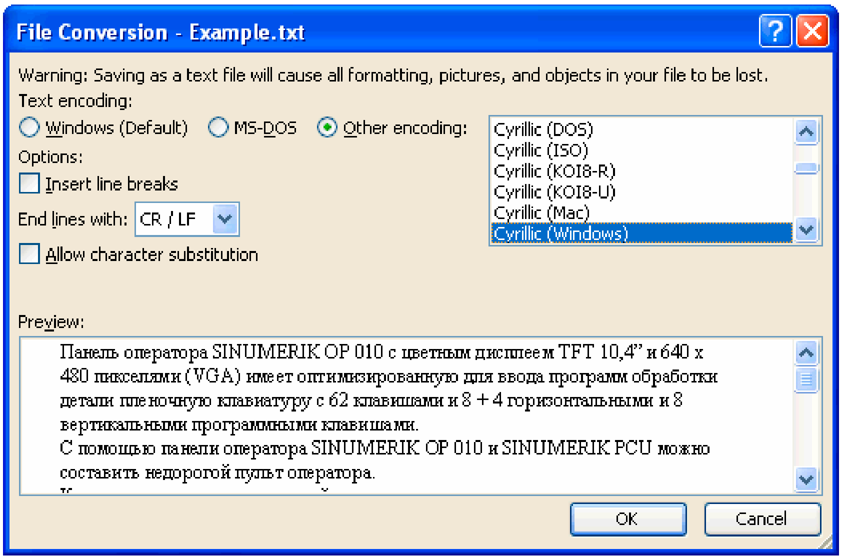
Save As

Save in: Examples

File name: Example.txt

Save as type: Plain Text (*.txt)

2. Choose the option "Text Only (*.txt)" under "Save as type".
3. Enter the name of the file in the "File name" box and click "Save".
   – The "File Conversion" dialog box will then appear.



File Conversion - Example.txt

Warning: Saving as a text file will cause all formatting, pictures, and objects in your file to be lost.

Text encoding:

○ Windows (Default) ○ MS-DOS ● Other encoding:

Cyrillic (DOS)
Cyrillic (ISO)
Cyrillic (KOI8-R)
Cyrillic (KOI8-U)
Cyrillic (Mac)
Cyrillic (Windows)

Options:

□ Insert line breaks

End lines with: CR / LF

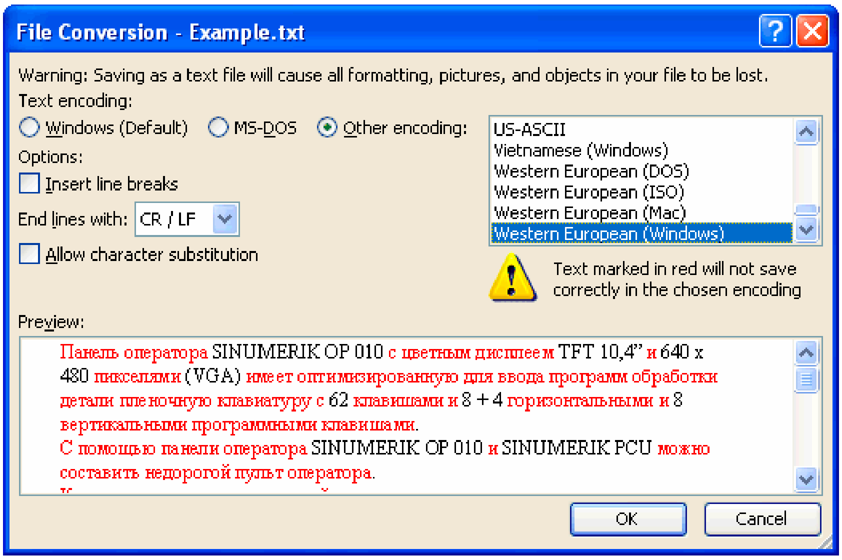
□ Allow character substitution

Preview:

Панель оператора SINUMERIK OP 010 с цветным дисплеем TFT 10,4" и 640 x 480 пикселями (VGA) имеет оптимизированную для ввода программ обработки детали пленочную клавиатуру с 62 клавишами и 8 + 4 горизонтальными и 8 вертикальными программными клавишами.
С помощью панели оператора SINUMERIK OP 010 и SINUMERIK PCU можно составить недорогой пульт оператора.

   – Select the encoding which is suitable for your target language and click "OK". Section "Codes for SINUMERIK HMI", Table 2-1 lists the suitable encoding options.
   – Your file is now saved as a text file and encoded using the selected code page in the process.

- If you have selected a completely unsuitable code then a warning is displayed in the dialog box. This warning will also appear if you have selected the correct code but your file contains characters which are not permitted, such as Latin mutated vowels in a Cyrillic text.



File Conversion - Example.txt

Warning: Saving as a text file will cause all formatting, pictures, and objects in your file to be lost.

Text encoding:

○ Windows (Default) ○ MS-DOS ● Other encoding:

US-ASCII
Vietnamese (Windows)
Western European (DOS)
Western European (ISO)
Western European (Mac)
Western European (Windows)

Options:

□ Insert line breaks

End lines with: CR / LF

□ Allow character substitution

Text marked in red will not save correctly in the chosen encoding

Preview:

Панель оператора SINUMERIK OP 010 с цветным дисплеем TFT 10,4" и 640 x 480 пикселями (VGA) имеет оптимизированную для ввода программ обработки детали пленочную клавиатуру с 62 клавишами и 8 + 4 горизонтальными и 8 вертикальными программными клавишами.
С помощью панели оператора SINUMERIK OP 010 и SINUMERIK PCU можно составить недорогой пульт оператора.

- If you choose to ignore the warning, then all characters which cannot be encoded are replaced with a question mark.

2.5 Codes for SINUMERIK HMI

When you open or save a file, code page names are used rather than code page numbers, see Section "Opening an existing file" or "Saving text files".

Depending on the installation of your system, many other codes may be listed, including those for Unix (ISO), Apple (Mac), OEM (DOS) and IBM (EBCDIC) platforms.

Please use the Windows (ANSI) codes for SINUMERIK HMI.

Use the following codes to open and save files (shown in alphabetical sequence in accordance with the Word listing):

Table 2-1 Encoding

Description	Code page
Traditional Chinese (Big5)	950
Simplified Chinese (GB2312)	936
Japanese (Shift-JIS)	932
Korean	949
Cyrillic (Windows)	1251
Central European (Windows)	1250
Turkish (Windows)	1254
Western European (Windows)	1252

Input of Text 3

3.1 Special characters in different language families

Latin language family

With the Latin-based language families (Baltic, Central European, Turkish, Western), you can enter most of the characters, i.e., the basic Latin alphabet, directly with your (e.g., Western) keyboard.

Most of the special characters, such as ĄáâåçđéèÍïŁŇñöőÓťŤůū, are made up of the Latin letters A-Z a-z and diacritical characters such as acute ', breve ˘, cedilla, circumflex (caret) ˆ, point ˙, colon (dieresis, trema) ¨, double-acute ˝, grave accent ', háček (caron) ˇ, macron ¯, ogonek (nasal hook, crooked hook) ˛, squiggle ˚, slash /, hyphen – or tilde ˜.

Other characters of the alphabet are special letters (Þþ, Đð, ß) or ligatures (Ææ, Œœ, IJij). These are supplemented by language-specific abbreviations and punctuation marks, e.g., for the endings of the ordinal numbers in Spanish (ª and º) as well as the upside-down exclamation and interrogation marks (¡ and ¿).

The input methods described in the following will apply without restrictions to the Latin-based language families.

Greek/Cyrillic language family

Certain special considerations need to be taken into account when inputting texts using Greek or Cyrillic fonts.

Read Section "Special considerations when working with Greek and Cyrillic texts" for additional information.

East Asian languages

Certain special considerations also need to be taken into account when generating texts in Japanese, Chinese or Korean.

Read Section "Special considerations when working with East Asian texts" for additional information.

Disruptive Word options

Deactivate the following options in Word to prevent unwanted characters from reaching the text:

In the dialog box "Tools" → "AutoCorrect Options" go to the "AutoFormat as you type" tab and under "Replace as you type" check:

- "Straight quotes" with "Smart quotes"
- English ordinals (1st) with superscript
- Fractions (1/2) with fraction character (½)
- Special characters (--) with symbols (—)

"Apply as you type":

- Automatic bulleted lists
- Automatic numbered lists

If you do leave any of the options active then you can always use Ctrl+Z to undo any automatic corrections. As a general rule this will give the same results as if the option was deactivated.

3.2 Inputting characters via the Character Map

Note

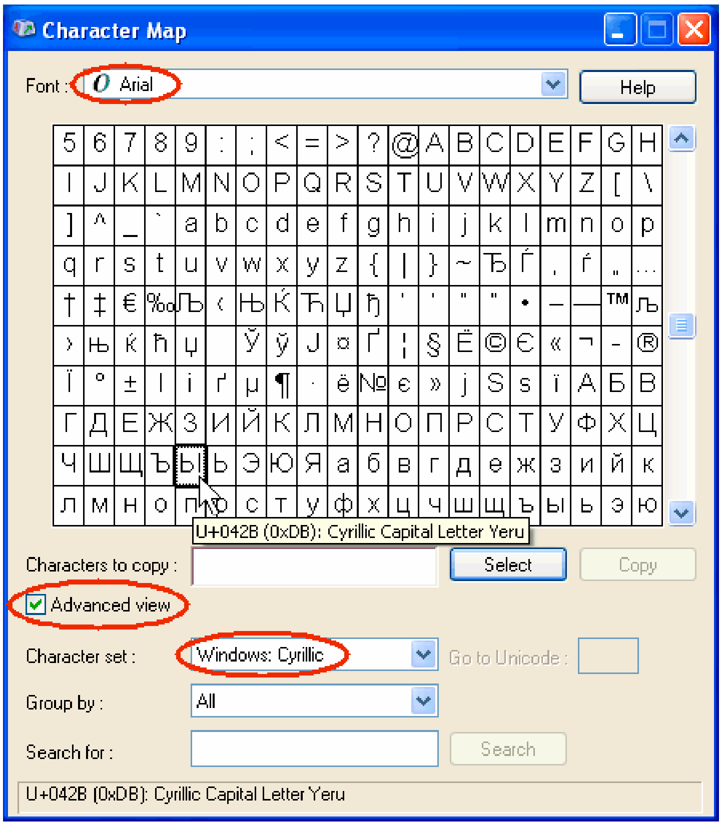
The Character Map integrated in Word (menu commands "Insert" → "Symbol...") is not suitable for generating code page-encoded text files, as it cannot be limited to the relevant code page.

Use the "Character Map" system program as described below.

Selecting the Character Map

1. In the start menu:

   select "Programs" → "Accessories" → "System Tools" → "Character Map" to start the "Character Map" tool.
2. From "Font", choose the same font type as you have selected for the text file.



Font : Arial

U+042B (0xDB): Cyrillic Capital Letter Yeru

Characters to copy :

☑ Advanced view

Character set : Windows: Cyrillic

Go to Unicode :

Group by : All

Search for :

U+042B (0xDB): Cyrillic Capital Letter Yeru

3. Activate the box "Advanced view".
4. Under "Character set" select the code page for your text file in order to restrict the number of available characters to this code page.

   The code pages are referred to as follows:

Table 3-1 Text file coding

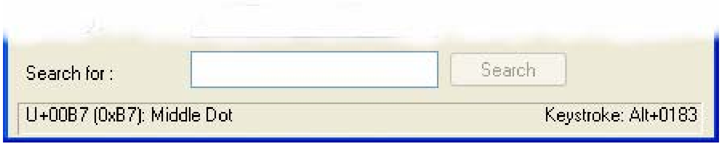
Description	Code page
Windows: Chinese (Taiwan)	950
Windows: Chinese (PROC)	936
Windows: Japanese	932
Windows: Korean	949
Windows: Cyrillic	1251
Windows: Eastern European	1250
Windows: Turkish	1254
Windows: Western	1252

Inputting characters

1. Choose a character you want to input.

   The information bar at the bottom will display the Unicode code (U+..., hexadecimal), the code page code (in brackets) and the clear text description of the character.

   If you have set up the keyboard assignment accordingly (see Section "Using a different keyboard assignment"), the keystroke combination for entering the character numerically will appear at the bottom right in the status bar (see Section "Numerical text input").



U+00B7 (0xB7): Middle Dot

Keystroke: Alt+0183

2. Click "Select".

   The character is copied into the "Characters to copy" field.
3. Repeat steps 1 and 2 for further characters.
4. After you have selected all the characters, click "Copy".

   The characters are copied from the "Characters to copy" field to the clipboard.
5. Switch back to Word and press Ctrl-V.

   The character is copied from the clipboard to the text.

3.3 Compound input

With many of the compound special characters it is possible to input the diacritical character first and then the letter. The diacritical character will not appear at first and is then combined with the letter to form a special character.

Example

With a German keyboard assignment, press, for example, the keys ' (acute) and "a" one after the other to obtain the character á (a-acute).

The keys with the diacritical characters, which are thus dealt with in a special way, are marked with a special color in the descriptions of the keyboard assignments.

Keyboard assignments

The table below shows some keyboard assignments for the supported diacritical characters:

Keyboard assignments	Acute	Grave	Circumflex	Háček	Breve	Ogonek	Point	Colon	Cedilla	Double acute	Squiggle	Tilde
	'	'	ˆ	ˇ	˘	˛	˙	¨	¸	˝	˚	˜
German	X	X	X									
French		X	X					X				X
Polish	X		X	X	X	X	X	X	X	X	X	
Spanish	X	X	X					X				X
Czech	X		X	X	X	X	X	X	X	X	X	
Hungarian	X		X	X	X	X		X	X	X	X	
U.S. International	X	X	X					X				X

Obtain a graphical representation of your keyboard assignment to learn which key combinations produce the diacritical characters.

Keyboard assignment in picture form

Images of the keyboard assignments can be obtained from Microsoft via the Internet at the following address:

http://www.microsoft.com/globaldev/reference/keyboards.mspx

If you press the space bar after the diacritical character, you will see the character itself.

Note

It can be useful to switch the keyboard assignment to "U.S. International" which corresponds mainly to the American keyboard, but additionally contains 5 diacritical characters and provides many special characters.

3.4 Using a different keyboard assignment

Necessity of changing the keyboard assignment

When is a change in the keyboard assignment necessary?

- If you have to input a lot of text.
- Your translator, who is used to a certain language-specific keyboard, wants to use your PC.

However, if you (as a German) only need to make smaller changes to existing files, then the German keyboard assignment may be enough, provided there are only a few special characters which cannot be represented with the German keyboard. You can enter these characters using the Character Map.

Example

In Hungarian, for example, you would have to input these characters as follows:

- őŐűŰ (O double-acute, U double-acute) via the Character Map
- öÖüÜ directly
- áÁéÉíÍóÓúÚ compound input

3.4.1 Concept of the keyboard language

Windows manages pairs for the keyboard consisting of "Language ("Language and Region Support")" and "Keyboard assignment". The set keyboard language determines which language property is assigned to the characters entered. The associated keyboard assignment will determine which key produces which character.

Several such settings can be set up and it is possible to switch between them. One of the settings is defined as the default setting.

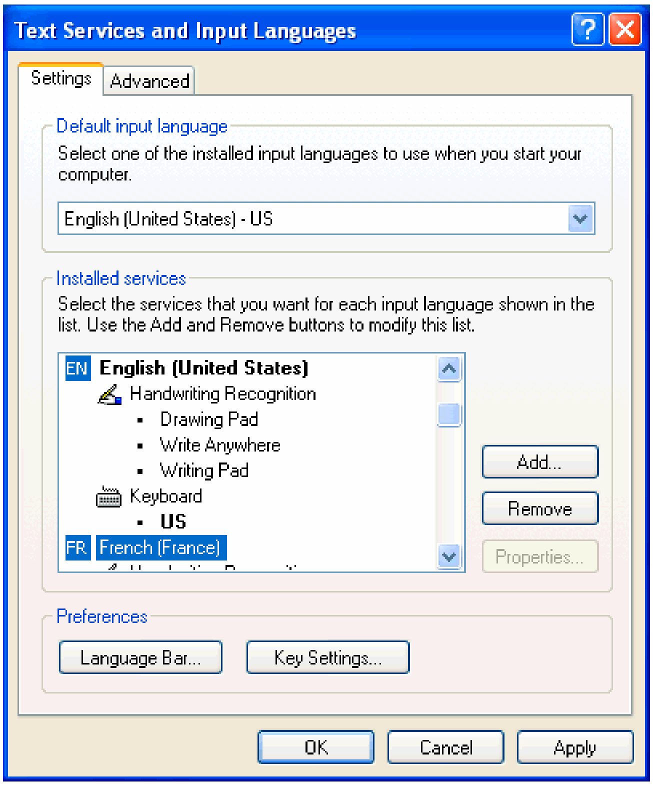
Windows uses the term "Language and Region Support" as this can be used not only to switch the language properties, but also to select settings such as currency, date format etc.

3.4.2 Setting up a keyboard assignment in Windows XP

Proceed as follows to select a new keyboard assignment:

1. From the Start menu, select "Settings" → "Control Panel" and click "Regional and Language Options".
2. Click the "Languages" tab and click the button "Details..." under the heading "Text Services and Input Languages".

   If a symbol such as DE is displayed in the right-hand taskbar, you can also right-click this symbol and select the command "Properties..." to speed up steps 1 and 2.
3. The dialog box "Text Services and Input Languages" is opened. Select the "Settings" tab.



Default input language

Select one of the installed input languages to use when you start your computer.

English (United States) - US

Installed services

Select the services that you want for each input language shown in the list. Use the Add and Remove buttons to modify this list.

EN English (United States)
Handwriting Recognition
- Drawing Pad
- Write Anywhere
- Writing Pad

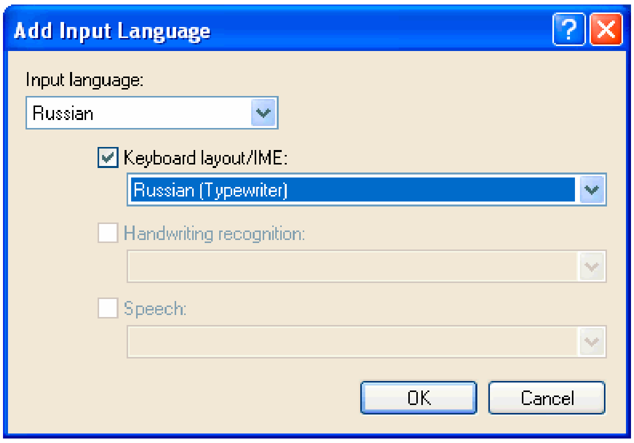
Keyboard
- US

FR French (France)

Preferences

4. Click "Add".

   The dialog box "Add Language and Region Support" is opened.



Input language: Russian

☑ Keyboard layout/IME: Russian (Typewriter)

□ Handwriting recognition:

□ Speech:

5. In the dialog box "Add Language and Region Support", select a language under "Language and Region Support" and the associated keyboard assignment under "Keyboard Assignment".

   The language ("Language and Region Support") may also be the same as the default language or as some other language for which you have already set up a keyboard assignment. However, in this case you will not be able to make such a good distinction between the two keyboard assignments, as Windows usually only displays the language ("Language and Region Support").

6. Click "OK". The dialog box "Add Language and Region Support" is closed. You then return to the "Text Services and Input Languages" window.
7. Use "Keyboard…" to select the way in which you want to switch the keyboard assignment. If you select "none", you can only switch using the mouse.
8. Under the "Language and Region Support" bar you can select whether and how the current keyboard assignment is displayed in the taskbar or in a special button bar.
9. Click "OK" to quit the dialog box.

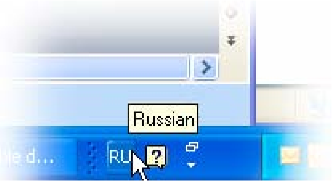
3.4.3 Use

You can now create text with a "keyboard assignment" suitable for the target language (for the input of your texts in a foreign language) **or** with your previous default keyboard assignment (for the remaining operation). You can switch between the two assignments as required.

Now, a small blue field with a two-letter language abbreviation should appear in the taskbar for the language you have set:

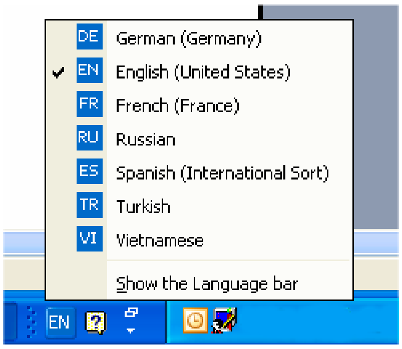
If you position the mouse pointer on this field (without clicking it), the setting is displayed.

If the selected keyboard assignment is the default assignment for the selected language (e.g., "Turkish" – "Turkish Q"), then only the language ("Turkish") is displayed. Otherwise, the complete settings with language and keyboard assignment will be displayed ("Turkish – Turkish F").

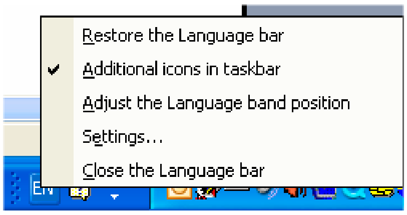


If you click this field with the **left** mouse button, the languages you have set will be offered in a small menu.

The current setting is marked by a check mark.



If you click this field with the **right** mouse button, an abbreviation of the dialog box "Keyboard Properties" is displayed above the "Properties" menu option.


Restore the Language bar
✓ Additional icons in taskbar
Adjust the Language band position
Settings...
Close the Language bar
EN

Among further settings that are offered is the ability to arrange this display as an Input language bar independently of the taskbar: at the top edge of the screen, for example.

Keyboard assignment specific to input window

The Windows operating system will memorize the current keyboard assignment **for each input window.**

- The keyboard assignment is switched automatically if you switch to another window.

  A new window is always started with the default setting. After starting Word, you may have to reselect the keyboard assignment.
- Enable the window in which you want to input texts before you select the keyboard settings for this window.

Menu operation

All menus in the active program will also use the changed keyboard assignment. You will possibly no longer find certain key combinations with Alt and the letter key or Ctrl with a letter key, or even initiate a wrong command by mistake. You should therefore use the mouse or, in menus, the arrow keys.

Note

If you do not want to switch back to the other keyboard assignment quite so frequently, you can also declare the new combination with the special keyboard assignment to be the default target language in the "Text Services and Input Languages" dialog box by selecting Start menu "Settings" → "Control Panel" → "Regional and Language Options" → "Languages" tab:

Select the new setting under "Default Language and Region Support".

Please note that all new windows, the Start menu and the desktop will now also use the new keyboard assignment.

Keyboard assignment in picture form

You can obtain the keyboard assignment in picture form.

For appropriate Internet addresses, see Section "References for text generation"

Checking the keyboard assignment

Compare the keyboard assignment and note a key with two different assignments. If you are not sure, press this key when inputting the text to test which assignment is active.

Example: If the German keyboard assignment is active, pressing key "1" (letter group) will produce a "1" and if the Czech keyboard assignment is active, a "+".

Note

Check whether the automatic option for switching over the keyboard has been activated in Word. Inadvertent switching of the keyboard assignment by Word can cause confusion.

Section "Multi-language Word files" describes situations in which the automatic keyboard switching function is useful and how to activate and deactivate it.

3.5 Numerical text input

You can only use numerical input with Word if you have selected a suitable keyboard assignment (see Section "Using a different keyboard assignment").

Press the ALT key, and while holding down the key, enter zero and then the three-digit decimal coding of the desired special character on the numerical key group. Only then release the ALT key.

Example (Turkish):

ALT+0222 results in Ş (S with cedilla).

To produce this character, you will only need the Character Map of the appropriate code page (see Section "Language families").

Section "Input via the Character Map" also describes where the key combination is displayed in the "Character Map" tool.

Overview tables of the code pages

You will find overview tables of the code pages at, e.g., Microsoft under:

http://www.microsoft.com/globaldev/reference/WinCP.mspx

From these tables, combine the headers for the column and row (hexadecimal 00 to FF) and convert this number to form a decimal number (0 to 255).

The 4-digit hexadecimal numbers contained in the individual table fields are the corresponding Unicode codes which can be used for identification of a character, but they cannot be used for numerical input of that character.

3.6 Language properties in Word files

Language property

The language is a property (attribute) that is managed separately in Word for each character, in the same way as other text properties (bold, underline, etc.):

- In the language properties, Word will remember the language to which the character, the character sequence, the word or the whole sentence belongs.
- The language property is independent of the font. For example, a Cyrillic character can possess the language property "German".
- The language property is managed internally as a combination of main language and sublanguage, e.g., "French (Canada)".

Microsoft sometimes uses the designation "Region scheme" for the language property: this term also covers properties such as currency, date format, decimal separators etc.

Effects of the language property

This property is used (evaluated) for the spell checker and for automatic keyboard switching (see Section "Multi-language Word files").

If you do not wish to use either the spell checker or the automatic keyboard switching function, there is generally no need to worry about the language properties. In this case, do not forget to deactivate the automatic keyboard switching function.

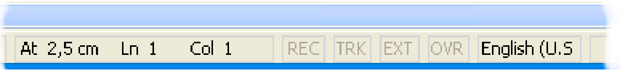
Saving the language property

Microsoft Word saves this property in Word files (*.doc). If a document is saved as a text file (*.txt) then this property is lost along with the other text properties.

If you wish to use the language property, it therefore makes sense to use a Word file to edit the texts and to keep it for subsequent changes. Whenever you reach a stage where you want to stop editing, you can save the document as a text file following the procedure described in Section "Saving text files".

Displaying the language property

In Word, the current language property is displayed in the status bar at the bottom edge of the window.



Setting the language property

Proceed as follows to set the language property for a text:

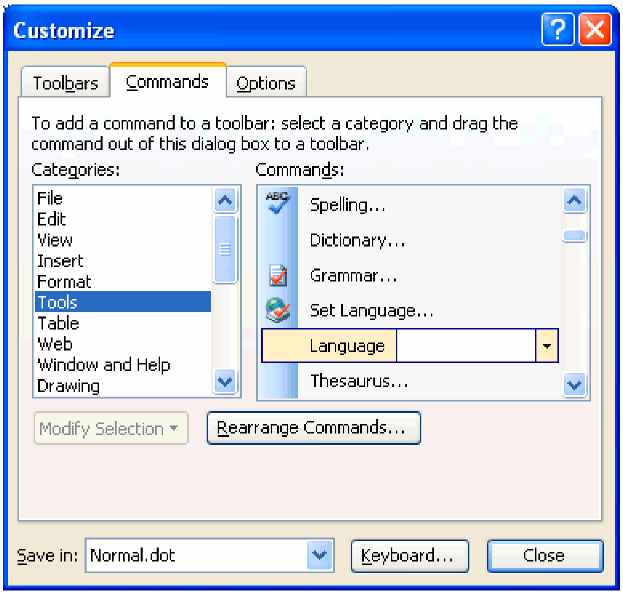
1. Highlight the part of the text for which you wish to set the language property, or press CTRL+A to select the entire text.
2. Then select the command "Extras" → "Language" → "Set Language...".
3. In the "Language" dialog box you can choose the correct language. Click "OK" when you are done.

Quick setting of the language property

In Word, you can insert a combined display and selection box into a symbol bar. This both displays the current language property and enables you to set the language as well.

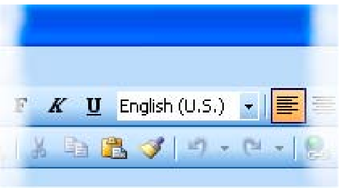
Proceed as follows to insert this field into a symbol bar:

1. In the "Extras" menu, select the command "Customize..."



To add a command to a toolbar: select a category and drag the command out of this dialog box to a toolbar.

2. In the "Customize" dialog box select the "Commands" tab.
3. Select "Tools" from the list of categories on the left.
4. Then select "Language" from the list of commands on the right (see Fig.).
5. Drag and drop this entry with the left mouse button into a symbol bar.



6. Close the dialog box.

With this selection box, you can then change the selected language for any part of the text.

Default setting of the language property

Word automatically sets the language property in the following situations:

- When opening a file of type "Encoded Text File" (see Section "Opening an existing text file").
- New text entered with the keyboard is automatically assigned the selected keyboard language.
- If the "Detect language automatically" option is selected, Word will try to assign the words entered to a language once they are complete and then automatically sets the language property.

Note

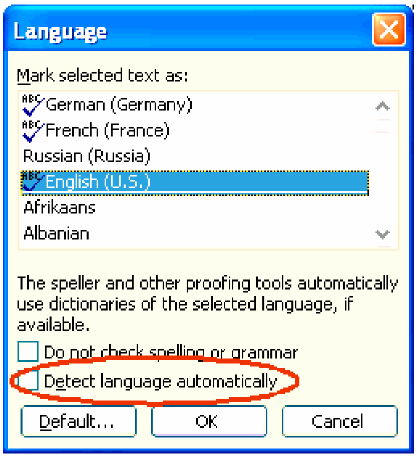
The "Detect language automatically" option can often deliver confusing results, particularly in technical texts.

This option is usually activated as a default setting.

Recommendation: Deactivate the "Detect language automatically" option.

Proceed as follows to deactivate the "Detect language automatically" option:

1. Make sure that no part of the text has been highlighted and select the command "Tools" → "Language" → "Set Language..."
2. In the "Language" dialog box, deactivate the option "Detect language automatically" and click "OK".



Language

Mark selected text as:

German (Germany)
French (France)
Russian (Russia)
English (U.S.)
Afrikaans
Albanian

The speller and other proofing tools automatically use dictionaries of the selected language, if available.

□ Do not check spelling or grammar
□ Detect language automatically

3.7 Multi-language Word files

In certain cases it may be useful to deliberately set different language properties for different parts of the text in a Word file.

Supplementary conditions for multi-language Word files

You want to enter certain parts of the text with the German keyboard assignment and then also edit these parts later on with the German keyboard assignment, whereas other parts of the text are to be generated and edited with the keyboard assignment of the target language.

Table with two language properties

To generate a multi-column text in which, for example, the first column possesses the language property "German" and the second column "Russian", for example, you can proceed as follows:

- Switch to the German keyboard and enter the first column of the first line. Then switch to the Russian keyboard and enter the second column:

  [Text 103][Mode 28] "йцукенгшщзхъфывапролджэячсмить"
- This does not need to be a Word table with borders and fields. You can also use a simple line and separate the different parts with spaces.
- Select the whole line, copy this to the clipboard by using Ctrl-C and paste this line several times by pressing Ctrl-V. This will give you two columns with German on the left and Russian on the right.

[Text 103][Mode 28]	"йцукенгшщзхъфывапролджэячсмить"
[Text 103][Mode 28]	"йцукенгшщзхъфывапролджэячсмить"
[Text 103][Mode 28]	"йцукенгшщзхъфывапролджэячсмить"
[Text 103][Mode 28]	"йцукенгшщзхъфывапролджэячсмить"
[Text 103][Mode 28]	"йцукенгшщзхъфывапролджэячсмить"
[Text 103][Mode 28]	"йцукенгшщзхъфывапролджэячсмить"

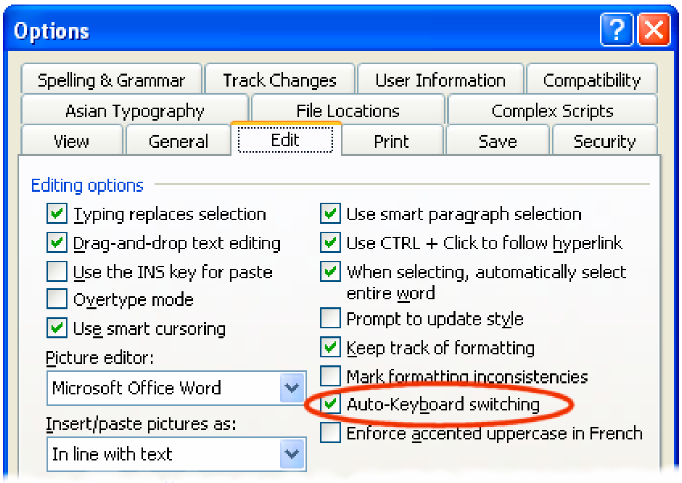
Examples

This can be necessary in the following cases:

- With Greek and Cyrillic texts, the Latin letters and other ASCII characters are missing in the appropriate keyboard assignments.
- With some keyboard assignments such as Czech, no access to the digits is granted on the standard keyboard level.
- The arrangement of the Latin letters deviates substantially from the arrangement to which you are accustomed (e.g., Turkish-F keyboard assignment).
- You, as a German, want to prepare one part of the text, and a second part is to be generated by your translator.

Automatic keyboard switching

In order to automatically have the correct keyboard assignment when working on different parts of the text, in Word go to "Tools" → "Options" and activate the "Auto-Keyboard switching" option in the "Edit" tab.



Editing options

☑ Typing replaces selection
☑ Drag-and-drop text editing
□ Use the INS key for paste
□ Overtype mode
☑ Use smart cursoring

Picture editor: Microsoft Office Word

Insert/paste pictures as: In line with text

☑ Use smart paragraph selection
☑ Use CTRL + Click to follow hyperlink
☑ When selecting, automatically select entire word
□ Prompt to update style
☑ Keep track of formatting
□ Mark formatting inconsistencies
☑ Auto-Keyboard switching
□ Enforce accented uppercase in French

Note

Accidental use of the automatic keyboard switching function can be confusing. Only use this option when the aforementioned supplementary conditions are met.

Active keyboard assignment always visible

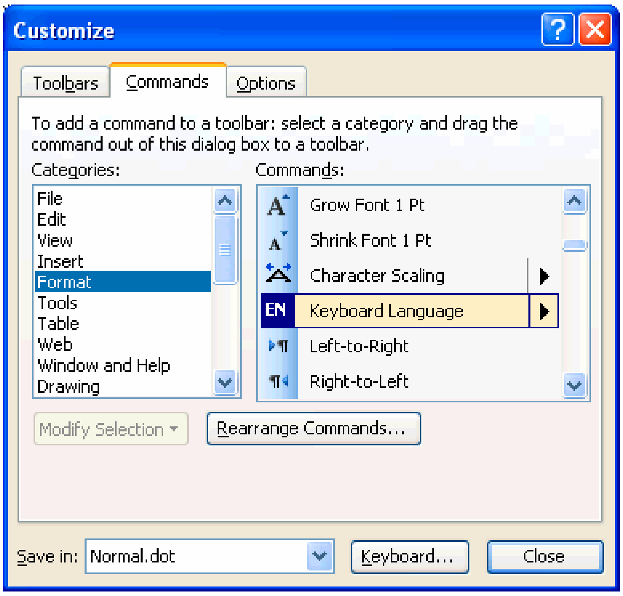
To make sure you know what is going on, you can adjust the taskbar to always display the currently active keyboard assignment.

1. Right-click the taskbar and select "Properties".
2. Activate the "Always keep the taskbar on top" option.
3. Deactivate the "Automatically hide taskbar" option.

You are also free to position the Language and Region Support bar anywhere you want on the screen. To do this, click the button for switching over the keyboard (e.g., **DE**) in the taskbar and select "Restore Language and Region Support Bar".

Another option is to activate keyboard switching in Word:

1. In the "Extras" menu select the command "Customize..."



To add a command to a toolbar: select a category and drag the command out of this dialog box to a toolbar.

Categories: Format

Commands: Keyboard Language

Save in: Normal.dot

2. In the "Customize" dialog box select the "Commands" tab.
3. Select "Format" from the list of categories on the left.
4. Then select "DE Keyboard Language" from the list of commands on the right (see Fig.).
5. Drag and drop this entry with the left mouse button into a symbol bar.

3.8 Special considerations when working with Greek and Cyrillic texts

Note

If you are generating texts in the Greek or Cyrillic language for the first time, then you MUST read this section!

From the point of view of the language sciences, the Greek and Cyrillic alphabets are considered as separate and independent alphabets, and not, for example, as an extension of the Latin alphabet.

This means that Greek and Cyrillic letters which, in their appearance, are identical to Latin letters, are not the same characters and may therefore also not be coded identically.

In some cases, this becomes evident by the fact that although certain letters look the same in upper case, the corresponding lower case letters look different.

Font	Name	Meaning
Latin	P, p	Latin letter P
Greek	Ρ, ρ	Greek letter RHO
Cyrillic	Р, р	Cyrillic letter ER
Latin	H, h	Latin letter H
Greek	Η, η	Greek letter ETA
Cyrillic	Н, н	Cyrillic letter EN

Example

Incorrect coding could have the following consequences:

Let us suppose that you wrote the Russian word for "NO" ("HET") using Latin letters instead of "НЕТ" using Cyrillic letters. Initially you would see no difference. However, a search for "нет" in lower case letters with the search option "Ignore uppercase/lowercase letters" would not find the incorrectly coded spelling.

Keyboard assignment for Greek/Cyrillic font

For this reason, the keyboard assignments for the Greek and Cyrillic fonts have been designed in such a way that the Latin letters are not available at all, not even via AltGr key combinations. These keyboard assignments have normally only two levels (standard and SHIFT).

Keyboard assignment for the Russian font

In the Russian keyboard assignment, there are also some ASCII special characters which are missing, such as #$[]{}"|~&.

For these languages it makes sense, therefore, to use Word files with two languages, as described in Section "Multi-language Word files".

Note

- Switch the keyboard to Latin only if this is absolutely necessary! When doing so, do not switch over for individual letters, but only for complete words and sentences.

  Never use words with mixed codings!
- Find out whether and to what extent certain German, English or international abbreviations (e.g. DIN/ISO) may or must be written using Latin fonts, or whether a conversion (ДИН/ИСО) is more appropriate, according to the conventions of the country in question.

  The key assignment for combinations, such as Ctrl-C, Ctrl-V, Ctrl-Z, Alt-A, etc. normally follow the US keyboard assignment.

3.9 Special considerations when working with East Asian texts.

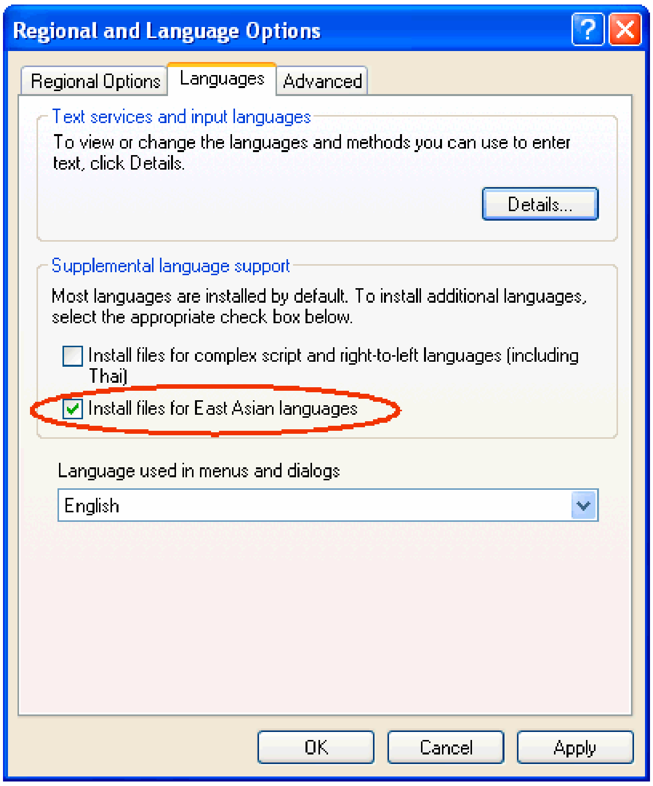
Requirement

To author or correct texts in Japanese, Chinese or Korean, you will need to have installed the corresponding expansion package for the operating system.

Checking the system installation

Open the dialog box "Regional and Language Options" via "Start" → "Settings" → "Control Panel".

Check the "Languages" tab to see whether the option "Install files for East Asian languages" has been selected.



Text services and input languages

To view or change the languages and methods you can use to enter text, click Details.

Supplemental language support

Most languages are installed by default. To install additional languages, select the appropriate check box below.

□ Install files for complex script and right-to-left languages (including Thai)

☑ Install files for East Asian languages

Language used in menus and dialogs: English

You will not usually need the installation CD to reinstall the files for East Asian languages in Windows XP, as the files are already present on the hard drive in compressed form.

You will, however, require Administrator Rights to perform the installation.

Input

An Input Method Editor (IME) is used for inputting. This is a tool which is used to put together the pictograms on a European keyboard and can be activated analogously to a keyboard assignment.

You will normally need a translator who speaks the target language as his/her mother tongue to operate the IME system.

There is no need to have a Windows operating system in the target language.

Full-width European characters

In the code pages for the East Asian languages, two versions of many of the European characters (Latin letters, numbers and the characters # $ % & etc.) are included:

- In the normal (narrow, half-width) version.

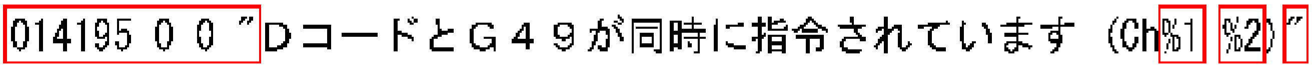
  This is the set of ASCII characters which is compatible with European code pages.

- In a double-width (full-width) version.

  These versions of the characters are twice as wide and are not compatible with the ASCII characters.

If you are using characters with a functional significance it is important to use the ASCII versions.

Correct example (SINUMERIK alarm text, Japanese):



014195 0 0 "Ｄコードと Ｇ４９が同時に指令されています（Ch%1 %2)"

The circled characters are:

014195 0 0	Identifier and attributes
Space character	Syntactical separator
Inverted commas	Text delimiters
%1, %2	Parameter variables

Only the ASCII versions must be used for these characters.

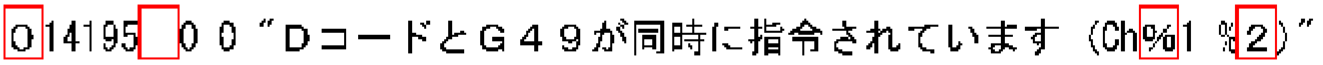
Note

In the remaining part of the text it is quite acceptable to use full-width characters for better readability (as in the example here for G49).

Notice

There is even a full-width version of the space character (ideographic space). You can only recognize this character from its width by selecting it.

Incorrect example:



Ｏ14195　0 0 “ＤコードとＧ４９が同時に指令されています (Ch％1 %２)”

The circled characters are full-width variants which have been used incorrectly.

Note

When placing a translation order, tell the translator that only the narrow, ASCII-compatible characters must be used for characters with a functional significance.

Simplified/traditional Chinese

Please make a careful distinction between simplified Chinese and traditional (or standard) Chinese. The former is used in the PR of China, whereas the latter is used in Taiwan. These two variants are coded differently and therefore have to be treated differently. You should therefore also make sure that this is clearly specified in your translation order.

Generating texts

When you are generating texts in Japanese, Chinese or Korean, it definitely makes sense to initially create a Word file.

There should be no problems if you are using Word 2003 to exchange documents with your external translator.

Afterwards, you can convert this Word file into a text file in the way described in Section "Saving text files".

4 Problems and remedies

Keyboard assignment switches unintentionally

Problem:

The keyboard occasionally switches to a different assignment while text is being entered, without this being explicitly requested.

Remedy:

Check whether the option for automatic keyboard switching is set as described in Section "Multi-language Word files". Deactivate this option.

Language properties are changed inadvertently

Problem:

After creating a text with certain keyboard assignments (e.g., Polish), you notice that parts of the text have German or English as the language property.

Remedy:

Deactivate the automatic language recognition option as described in Section "Language properties in Word files".

Hotkeys no longer work

Problem:

Certain commands (Ctrl+C, Alt+T etc.) no longer work.

Remedy:

As a result of a different keyboard assignment, the keyboard shortcuts for certain commands have changed. Use the mouse while you are working with the other keyboard assignment.

Individual texts not working

Problem:

In some cases, the finished text files do not work in the SINUMERIK Runtime system. Individual texts or lines of text are obviously not being read correctly.

Remedy:

Check the formal text criteria:

Are all of the characters which have a functional significance correctly encoded?

- If inverted commas are a requirement, check whether Word has converted the straight inverted commas (ASCII characters) that were entered into typographical inverted commas. Some disruptive Word options are listed in Section "Special characters in different language families".
- If the texts in question are East Asian texts, check the coding of the non-Asian characters. See Section "Special considerations when working with East Asian texts".

  If full-width characters have been used for functional characters, replace them with the corresponding ASCII variants.

Yen character (Japanese) or Won character (Korean) in the text

Problem:

The translator has obviously used a Yen character ¥ or a Won character W wherever you would expect a backslash “\”.

Remedy:

This is not necessarily a mistake. Instead, this has historic reasons. For many years the Yen character has been used in Japan and the Won character in Korea in place of the backslash character.

This representation was kept once the PC users in those countries had got used to using this representation in path names (where you would otherwise expect the backslash character). However, internally this is still the backslash character.

In the fonts "MS Gothic" (Japanese) and "Batang" (Korean) the backslash symbol is represented as a currency character.

However, it is also possible that the Yen character (Unicode U+00A5) or Won character (Unicode U+20A9) was actually entered instead of the backslash character (Unicode U+005C). To check this, proceed as follows:

1. Reformat the text in Word using a different font, e.g., "Courier New". If you see the backslash character in its normal representation, there are no problems.
2. However, if you can still see the currency symbol or a symbol which is not a valid symbol, you should correct the character and replace it with a backslash.
3. Afterwards, revert back to the normal font (MS Gothic or Batang).

References for text generation 5

For further information about generating texts, please refer to:

- Book: *Developing International Software, Second Edition*

  Microsoft Press, October 2002, ISBN 0-7356-1583-7

  http://www.microsoft.com/globaldev/getwr/dis_v2/default.mspx
- Overview tables of the Microsoft code pages

  http://www.microsoft.com/globaldev/reference/WinCP.mspx
- Pictures of the keyboard assignments in Microsoft can be found at the following address:

  http://www.microsoft.com/globaldev/reference/keyboards.mspx
- Windows XP FAQs:

  http://www.microsoft.com/globaldev/DrIntl/faqs/winxp.mspx
- Microsoft Global Software Development

  Detailed explanations on various aspects of international software

  http://www.microsoft.com/globaldev
- Unicode

  Tables, definitions, standards and tools

  http://www.unicode.org

Index

SIEMENS

SINUMERIK

SINUMERIK 840D sl PCU-Basesoftware (IM8)

Commissioning Manual

Valid for:

PCU-Basesoftware version 8.6

03/2009
6FC5397-1DP10-4BA0

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

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

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 adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the 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
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Ordernumber: 6FC5397-1DP10-4BA0
℗ 03/2009

Table of contents

Commissioning a system 1

1.1 Delivery condition of the system

Overview

The high-performance SINUMERIK PCU 50.3 has onboard interfaces for communicating via Ethernet, MPI and PROFIBUS DP. The integrated free slots remain free for other tasks. The PCU 50.3 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.
- For use with SINUMERIK 840D/840D sl:

  Two internal PCI slots are available for specific expansions.
- For use with SINUMERIK 840Di sl:

  A PCI slot is already equipped with the MCI2 board and can be equipped with the optional MCI board extension.

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 boot up.
- If the PCU is to be operated without an operator panel front, a monitor and an additional keyboard will also be required:
  - For diagnostics when booting 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 SP2	
Internet Explorer	V 6.0
MPI driver	V 6.03
Symantec Ghost (default)	V 8.2 (incl. Ghost Explorer)
TCU Support (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 Basesoftware, see the C:\BaseVers.txt file.

1.1.1 Hard disk partitions

Division of the hard disk

The hard disk has 40 GB of storage capacity and is divided into a primary partition C and an expanded partition with the three logical drives D, E and F which work with NTFS file access.

Depending on the order details, the HMI system software may be pre-loaded on delivery. It is installed subsequently by the customer the first time the system is booted up.

For reasons of data security, the HMI system software and the Windows XP system software are distributed on the different hard disk partitions.

The following figure shows how the hard disk of the PCU is used:

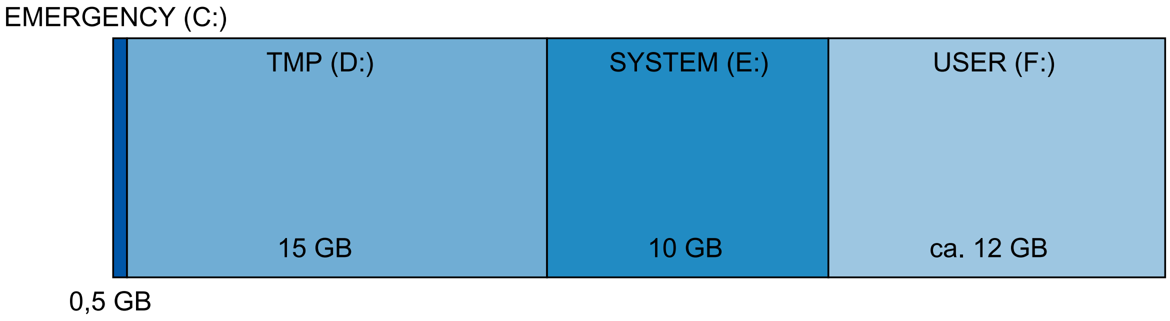


Figure 1-1 Division of the hard disk

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 status on delivery) and local backup images. 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. 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. Applications such as HMI system software (incl. data storage and temporary data), STEP7, OEM applications for HMI or customer-specific applications should only be installed here.

NOTICE

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 53)

Starting ServiceCenter Backup Restore (Page 63)

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:

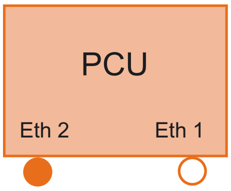
Windows services:	Start-up type:	
Computer Browser	Manual	(Not started)
Error Reporting Service	Disabled	
Portable Media Serial Number	Manual	(Not started)
SSDP Discovery Service	Disabled	
Universal Plug and Play Host	Disabled	
Web Client	Manual	(Not started)
Wireless Zero Configuration	Manual	(Not started)

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 50.350.3 has two Ethernet interfaces that are suitably preset with the SINUMERIK solution line for connecting to the system network.



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.

References: Operator Components and Networking Manual, Networking Chapter

See also

How to set the IP address of the PCU 50.3 (Page 34)

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). The 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).

- **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 terms of their main tasks:

User type	Task card
HMI (operator.group)	• Boot up of the PCU • Operating the HMI program • Windows Desktop
HMI (user.group)	• Boot up of the PCU • Operating the HMI program • Windows desktop
HMI+Service	• Service tasks • Boot up of the PCU • Operating the HMI program

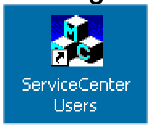
Service Desktop

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-up 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 Global user settings

"Global Settings"

As the **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

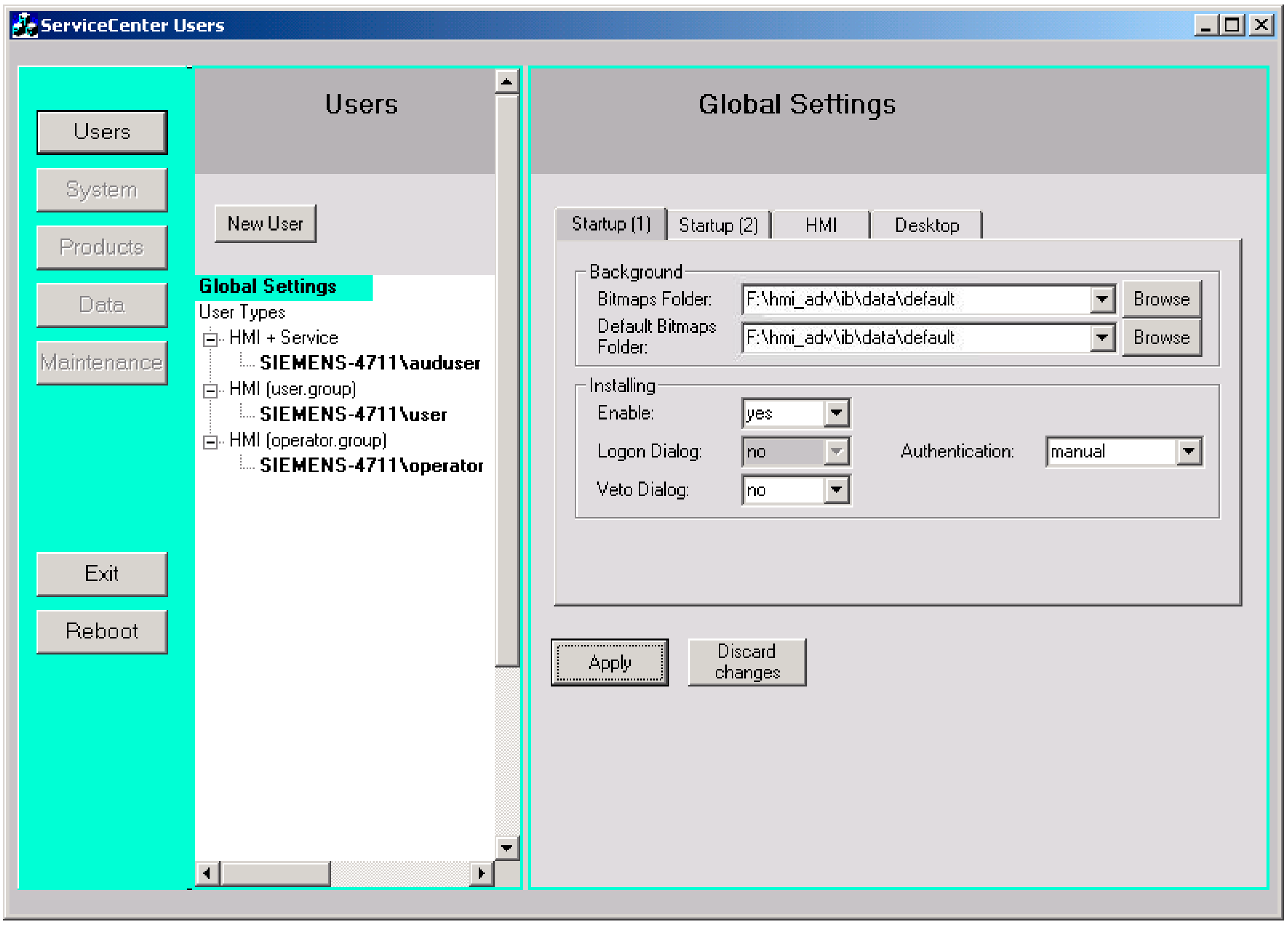
The scenario depicted in the images below assumes HMI-Advanced software has been installed.

Using ServiceCenter Users

Buttons:

- If you press the "Reboot" button, the system immediately reboots (without prior prompting).
- Use "Exit" to quit ServiceCenter Users.

"Startup (1)" tab



Bitmaps Folder: F:\hmi_adv\ib\data\default

Default Bitmaps Folder: F:\hmi_adv\ib\data\default

Enable: yes

Logon Dialog: no

Authentication: manual

Veto Dialog: no

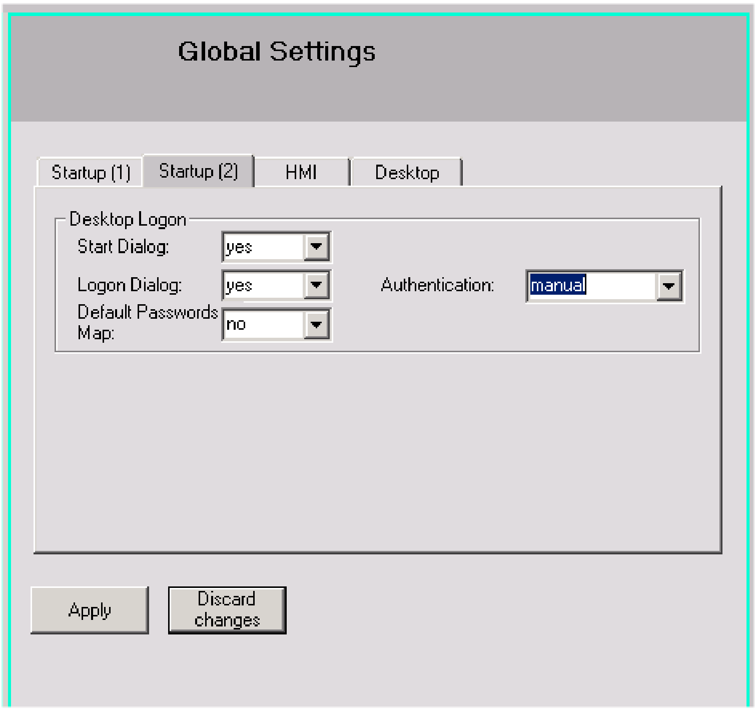
Figure 1-2 Global Settings: Startup (1) tab

Table 1- 1 The default is marked in "bold".

Startup (1) tab	Option	Effect
Background		
Bitmaps folder:	**F:\hmi_adv\ib\data\10700\0**	Directory with boot screens
Default bitmap folder:	**F:\hmi_adv\ib\data\default**	Directory containing default boot screens
Installing		
Enable:	**"yes"**	Authorization is in place to install user software during booting.
	"no"	No installation authorization
Logon dialog:	**"no"**	Display "Service Logon" dialog.
	"yes"	Do not display "Service Logon" dialog.
Veto Dialog:	**"no"**	For pending installation during the boot up: No prompting, the installation begins immediately.
	"yes"	For pending installation during the boot up: Prompt asking whether to install.

Startup (1) tab	Option	Effect
Authentication:	**"manual"**	Authentication of an HMI+Service user in the "Service Logon" dialog is undertaken manually.
	"EKS"	Authentication of an HMI+Service user in the "Service Logon" dialog involves EKS. The "Service Logon" dialog appears if authentication by EKS has failed in the background.
	"manual+EKS"	Authentication of an HMI+Service user in the "Service Logon" dialog either involves EKS or is performed manually.

"Startup (2)" tab


Global Settings

Startup (1) | Startup (2) | HMI | Desktop

Desktop Logon

Start Dialog: yes

Logon Dialog: yes

Default Passwords Map: no

Authentication: manual

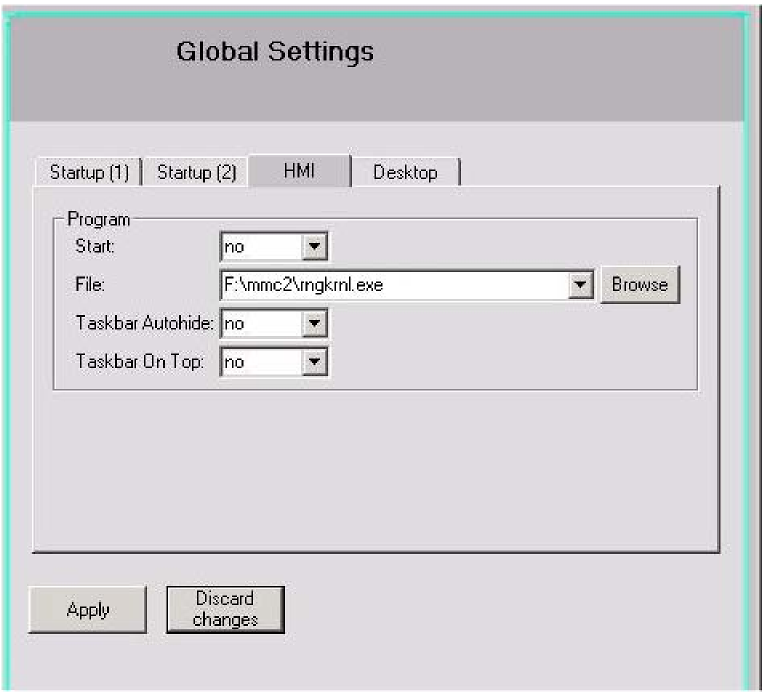
Figure 1-3 "Startup (2)" tab

Table 1- 2 The default is marked in "**bold**".

Startup (2) tab	Option	Effect
Desktop Logon		
Start dialog:	**"yes"**	Display "Desktop Access" dialog.
	"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.

Startup (2) tab	Option	Effect
	"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



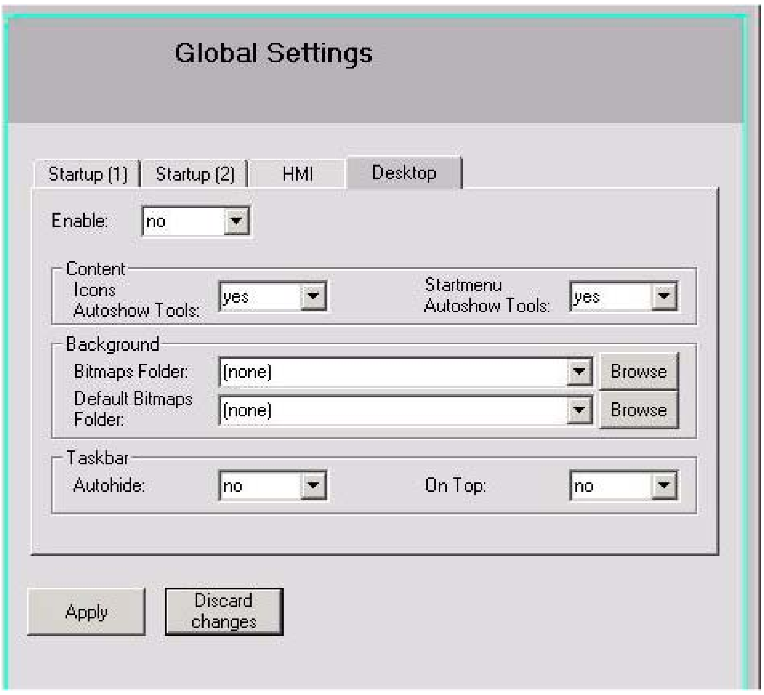
Start: no
File: F:\mmc2\rngkrnl.exe
Taskbar Autohide: no
Taskbar On Top: no

Figure 1-4 "HMI" tab

Table 1- 3 The default is marked in "**bold**".

"HMI" tab	Option	Effect
Program		
Start:	**"yes"**	The HMI program is started.
	"no"	The HMI program is not started.
File:	**F:\mmc2\rngkrnl.exe**	Directory of 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 bar always visible

"Desktop" tab



Global Settings

Enable: no

Content

Icons Autoshow Tools: yes

Startmenu Autoshow Tools: yes

Background

Bitmaps Folder: (none)

Default Bitmaps Folder: (none)

Taskbar

Autohide: no

On Top: no

Figure 1-5 Desktop tab

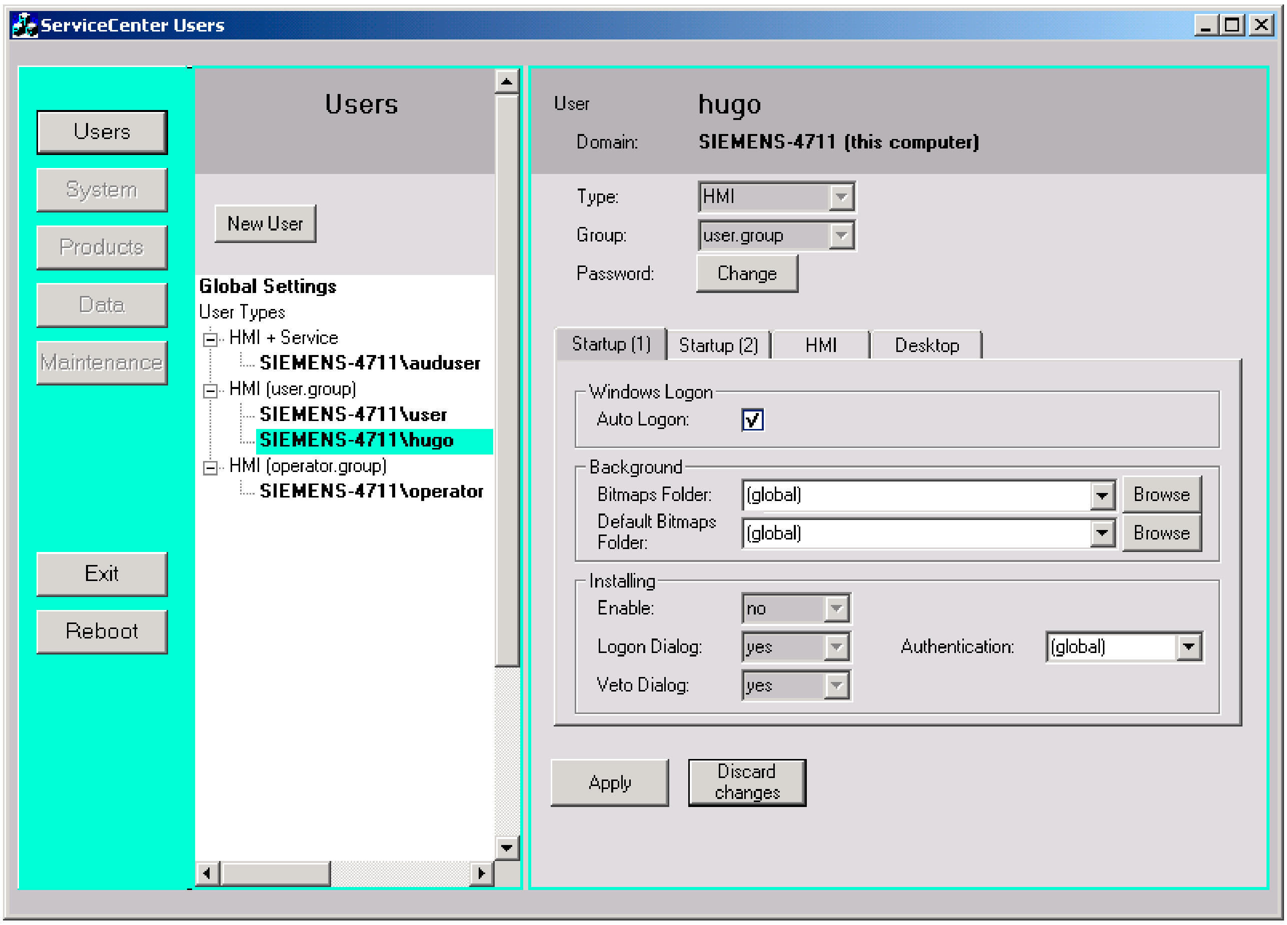
Table 1- 4 The default is marked in "bold".

"Desktop" tab	Option	Effect
Enable:	"no"	Dialog: Do not display "Desktop Access" dialog **(default setting: for HMI user)**
	"yes"	Dialog: Display "Desktop Access" dialog **(default setting: only for HMI+Service user)**
Content		
Icons Autoshow Tools:	"no"	Desktop: Start task bar in the background
	"yes"	Desktop: Start bar always visible
Startmenu Autoshow Tools:	"no"	Desktop: Start task bar in the background
	"yes"	Desktop: Start task bar always visible
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

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: "HMI (user.group)" user type



Global Settings
User Types
- HMI + Service
  - SIEMENS-4711\auduser
- HMI (user.group)
  - SIEMENS-4711\user
  - SIEMENS-4711\hugo
- HMI (operator.group)
  - SIEMENS-4711\operator

User hugo
Domain: SIEMENS-4711 (this computer)

Type: HMI
Group: user.group
Password:

Windows Logon
Auto Logon: ☑

Background
Bitmaps Folder: (global)
Default Bitmaps Folder: (global)

Installing
Enable: no
Logon Dialog: yes
Authentication: (global)
Veto Dialog: yes

Figure 1-6 Settings for the "HMI (user.group)" user

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.

1.2.3 How to create a new user

Overview

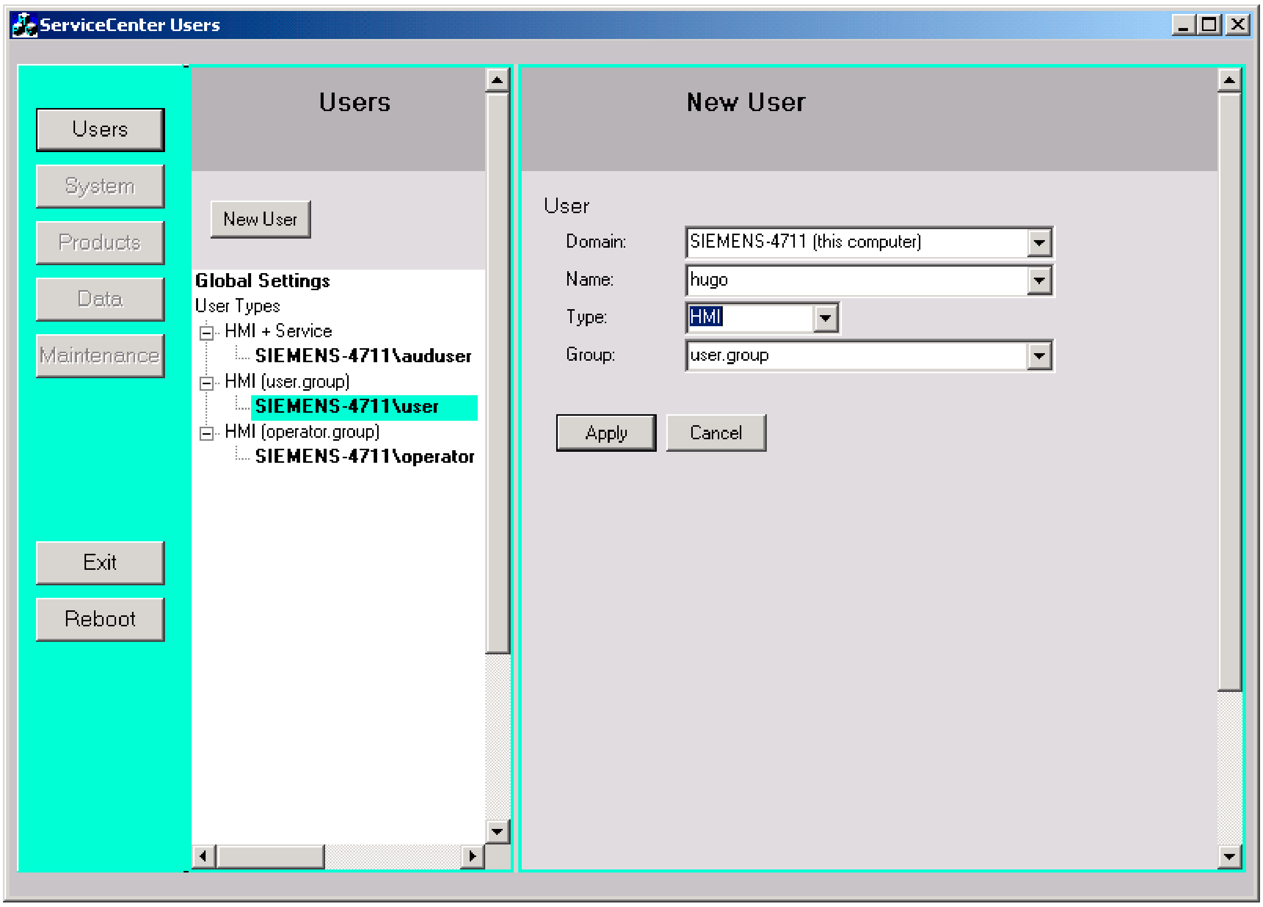
As an HMI+Service user, you can execute the following tasks in the Users ServiceCenter:

- Create new users.
- Delete users.
- Change user names.
- Add users from a domain.

Creating new users

To create a new user:

1. Click on "New User".
2. Enter a user name and assign a user type: e.g., "HMI + Service".
3. You will then be asked to specify a password.
4. Once you have confirmed by clicking on "OK", the new user will be set up and displayed in the list under "Users" (see following diagram).



Global Settings
User Types
HMI + Service
SIEMENS-4711\auduser
HMI (user.group)
SIEMENS-4711\user
HMI (operator.group)
SIEMENS-4711\operator

User
Domain: SIEMENS-4711 (this computer)
Name: hugo
Type: HMI
Group: user.group

Figure 1-7 ServiceCenter Users: New user

Deleting users

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 pop-up menu.

Result: The user is deleted with a confirmation prompt.

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.

Adding 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 "New User".
2. To do this, select the corresponding entry from the "Domain" list and a user from the "Name" list to assign the user to a user type on the PCU. The associated Windows user group is then assigned automatically.

See also

How to add the PCU to a domain (Page 35)

1.3 Boot up behavior of the PCU

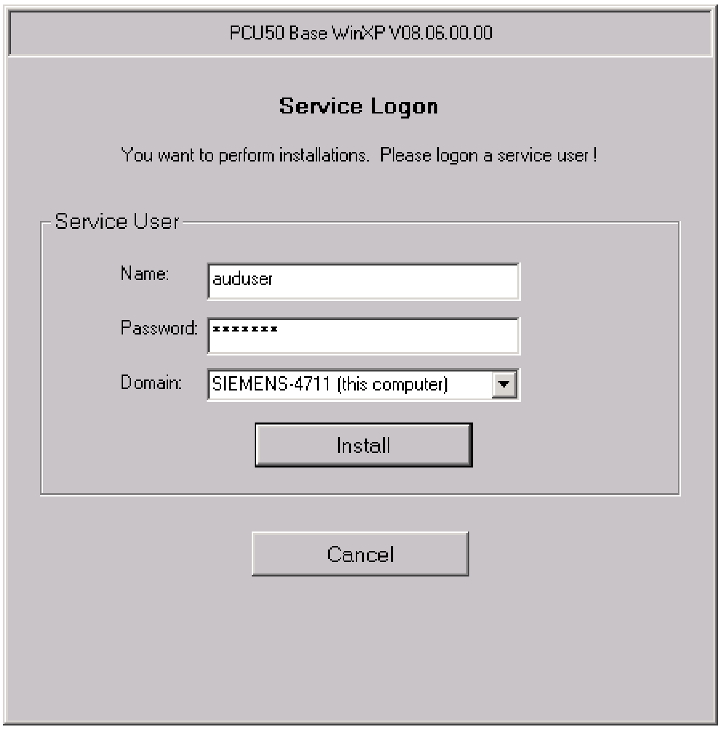
1.3.1 Service dialogs during boot up

Selection when booting up

The following dialogs are shown during boot up:

- "Service Logon" dialog

  This dialog is displayed if an "Install" directory is found on a bootable storage medium (e.g. USB-FlashDrive) when booting up.



PCU50 Base WinXP V08.06.00.00

Service Logon

You want to perform installations. Please logon a service user !

Service User

Name: auduser

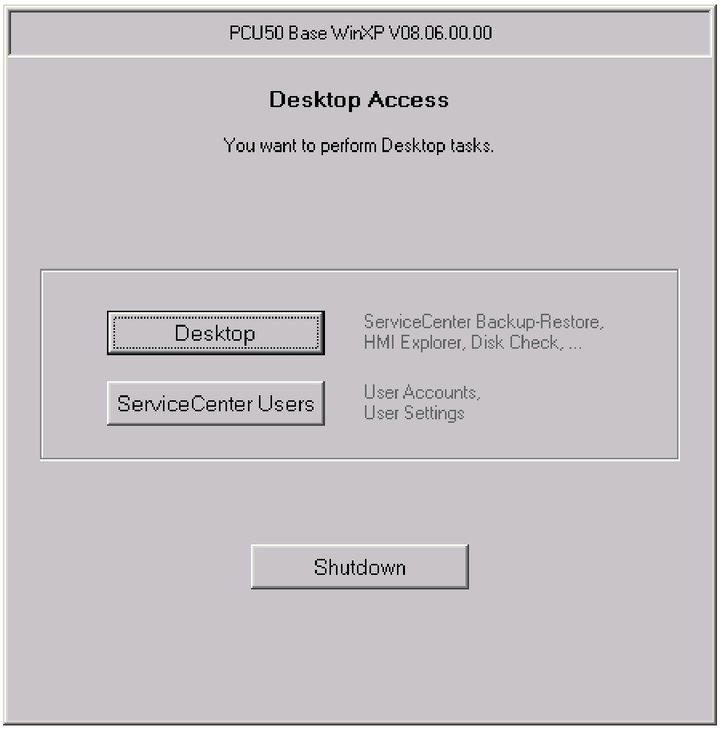
Password: *******

Domain: SIEMENS-4711 (this computer)

Figure 1-8 Service Logon

- "Desktop Access" dialog
  - This dialog is displayed **either** if no HMI program is installed **or** if HMI program startup is deactivated during booting.

    Displaying this dialog can be suppressed through configuration in the global or user settings. "Desktop" is then selected as the default.



PCU50 Base WinXP V08.06.00.00

Desktop Access

You want to perform Desktop tasks.

Figure 1-9 Desktop Access (example without HMI program)

Note

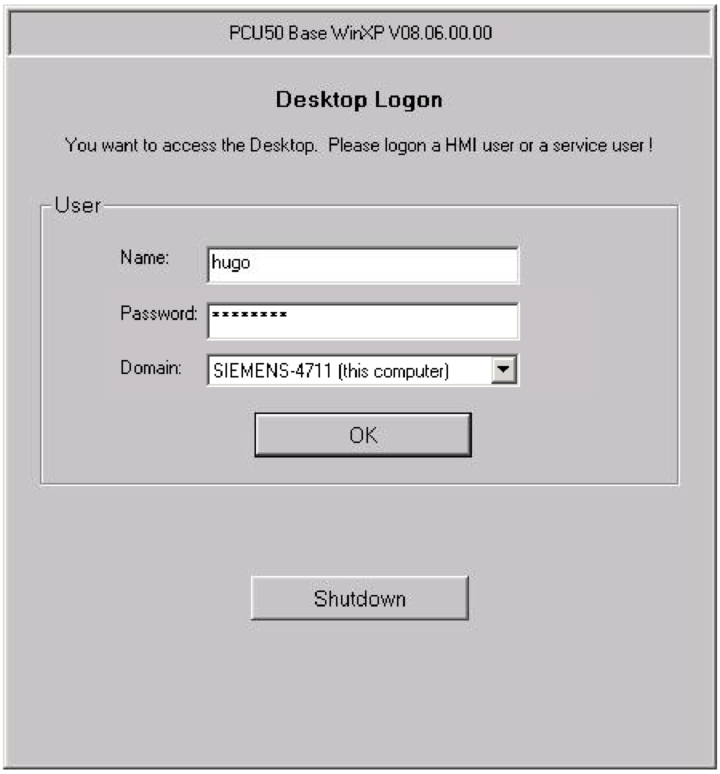
If an HMI program is installed, the "Start HMI" button is also available in the "Desktop Access" dialog.

- "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".



PCU50 Base WinXP V08.06.00.00

Desktop Logon

You want to access the Desktop. Please logon a HMI user or a service user !

User

Name: hugo

Password: ********

Domain: SIEMENS-4711 (this computer)

Figure 1-10 Desktop Logon (with authentication: "manual")

NOTICE

Setting the access level via EKS:

- 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

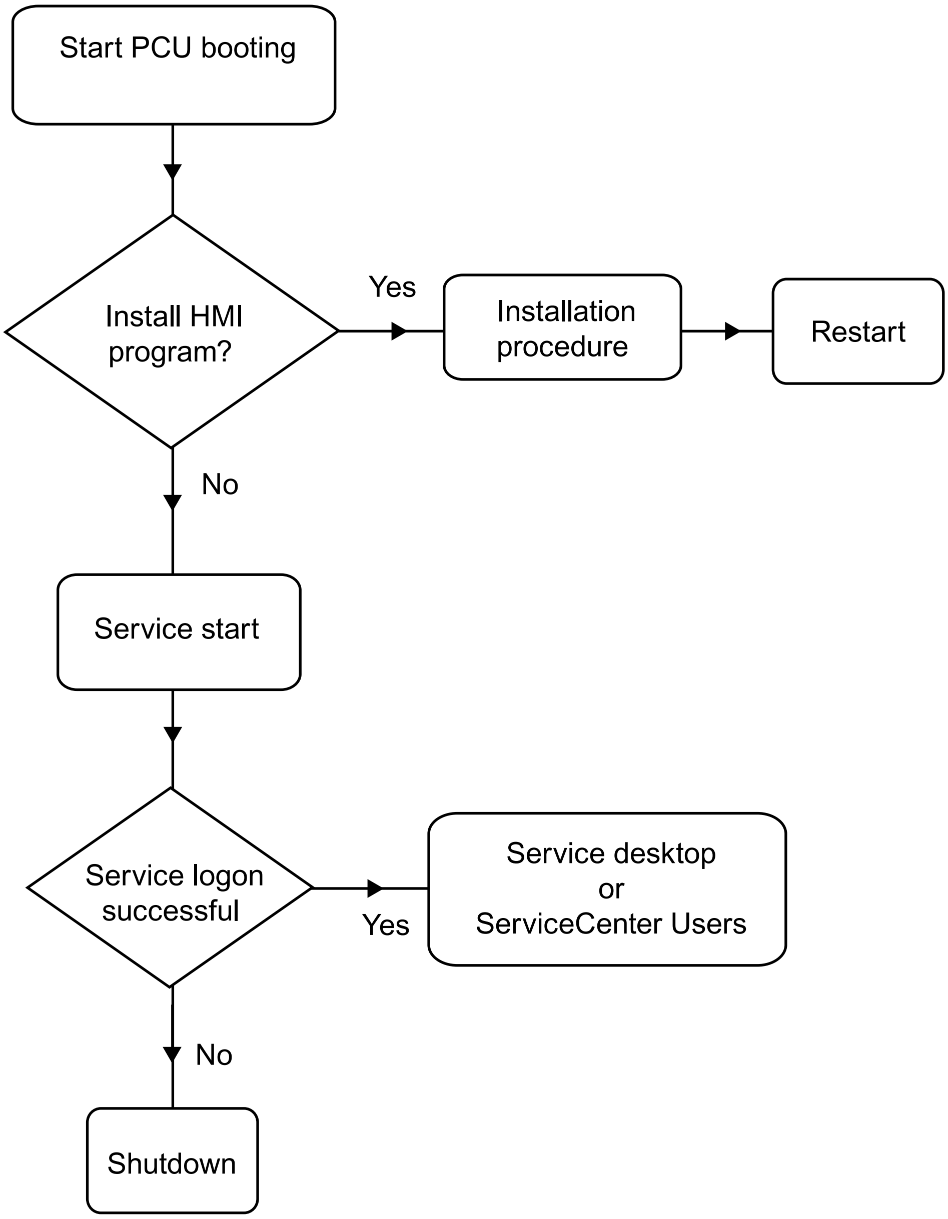


Figure 1-11 PCU bootup diagram (without HMI program)

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.
- "Service desktop", "ServiceCenter Users" or "Shutdown" can be selected in the service start dialog.
- Selecting "Service desktop" or "ServiceCenter Users" opens the service logon dialog.

Note

During the **initial** boot up of the PCU 50.3, 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

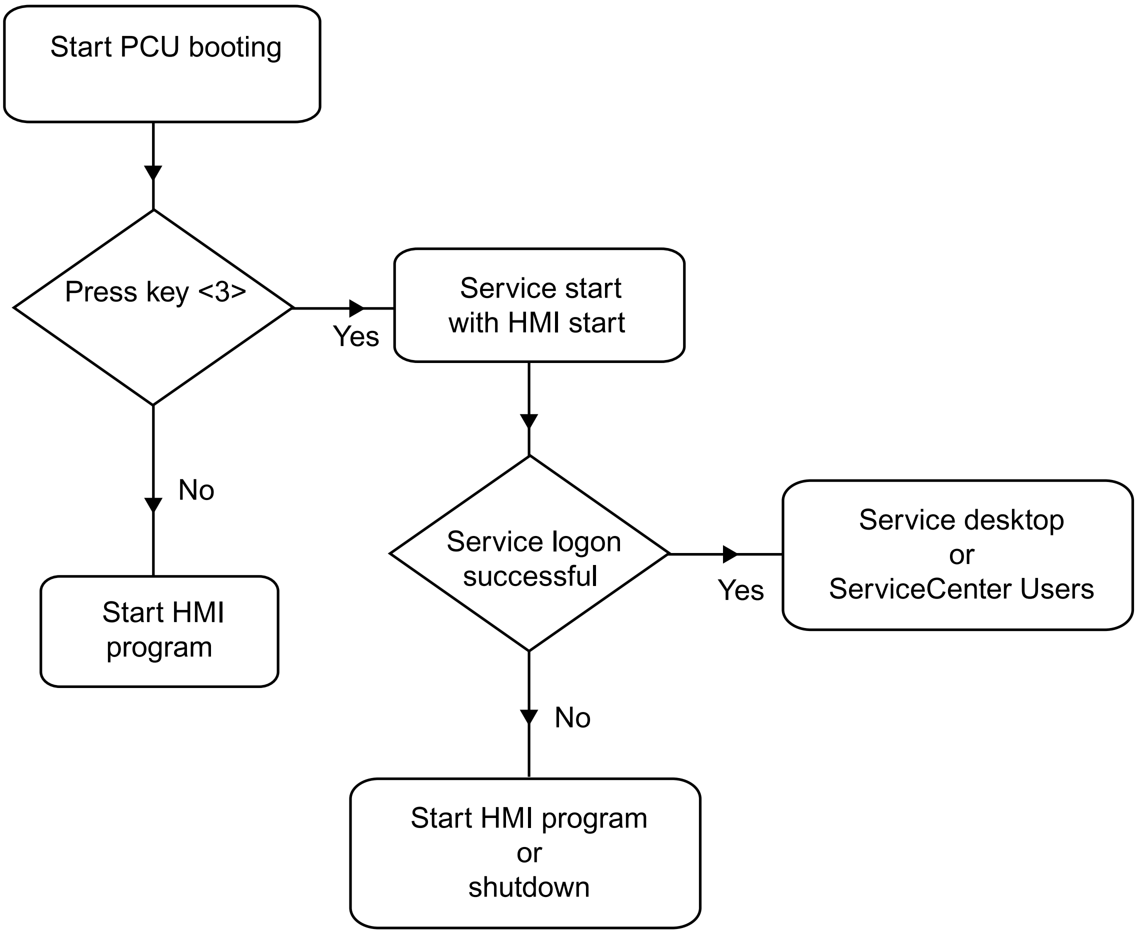

Start PCU booting
Press key <3>
Yes
Service start with HMI start
No
Start HMI program
Service logon successful
Yes
Service desktop or ServiceCenter Users
No
Start HMI program or shutdown

Figure 1-12 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. The service start dialog then 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 service 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: 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) [in active TCU mode: The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]: 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): [in active TCU mode: The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]: 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. (Different to AUTO_OP_2 !)
AUTO_OP_2	Like AUTO_OP_1, except: Example 2: There is no PCU panel, however there is a PCU monitor (irrespective of whether there are any TCU panels): [in active TCU mode: The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]: 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. 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) [in active TCU mode: The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]: 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. Example: In the case of a PCU monitor with a max. resolution of 1440x900, the SINUMERIK resolution setting is 1280x1024. 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): [in active TCU mode: The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]: 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. (Different to AUTO_MON_2 !)

Settings	Meaning
Example 3:	There is no PCU monitor and no PCU panel (= headless operation): A) in active TCU mode: a) dynamic resolution switching is enabled (resolution adaptation entry in TCU.ini) and at least one TCU is already logged on: The resolution is set to the resolution of the TCU which is currently active. 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: The resolution is set to the max. resolution of the current PCU panel, i.e., which logged on during the previous session. Default: Default TCU resolution in accordance with the registry. **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). A) in inactive TCU mode: 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: Example 2: There is no PCU monitor, however there is a PCU panel (irrespective of whether there are any TCU panels): [in active TCU mode: The following additional condition applies: PCU is activated (with dynamic resolution switching enabled)]: 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

Overview

The BIOS of the PCU 50.3 is preset in such a way that no changes are required. The date and time can be set under Windows or the HMI operator interface.

NOTICE
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:

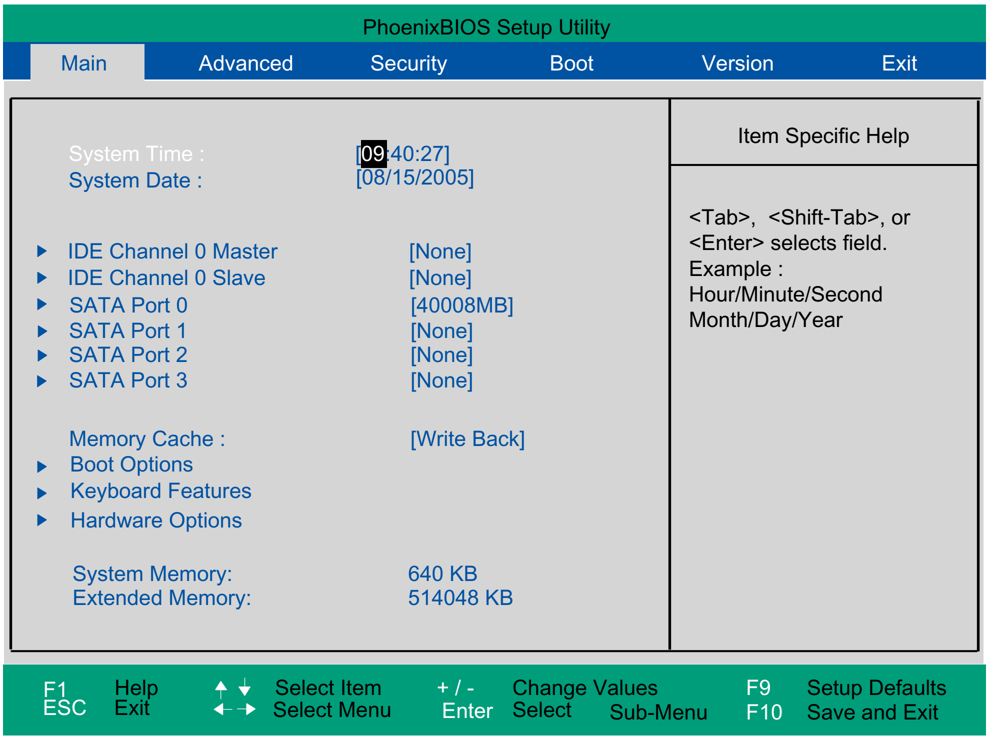
   Reset the device (warm or cold restart).

   After the first boot up, 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:



```
System Time :              [09:40:27]
System Date :              [08/15/2005]

▸ IDE Channel 0 Master       [None]
▸ IDE Channel 0 Slave        [None]
▸ SATA Port 0                [40008MB]
▸ SATA Port 1                [None]
▸ SATA Port 2                [None]
▸ SATA Port 3                [None]

  Memory Cache :             [Write Back]
▸ Boot Options
▸ Keyboard Features
▸ Hardware Options

  System Memory:             640 KB
  Extended Memory:           514048 KB
```

<Tab>, <Shift-Tab>, or <Enter> selects field.
Example :
Hour/Minute/Second
Month/Day/Year

Figure 1-13 BIOS Main Menu (Example)

BIOS setup: Defaults

The following system parameters are saved on delivery:

Menu: Main

System parameters	Defaults	Custom 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	Defaults	Custom 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	Defaults	Custom 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	Defaults	Custom 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	Defaults	
SIMATIC PC	SINUMERIK PCU50.3	
BIOS version	V05.01.06	
BIOS number	A5E00370214-ES005	
MPI/DP firmware	V01	
CPU type	Celeron ® M processor 1.50GHz	
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. Ramp up 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 in 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 powers up.

Note

Changes to the BIOS settings, with the exception of the boot sequence, require an OEM contract to be concluded.

Configuring the system 2

2.1 System settings

2.1.1 How to change the name of the PCU

Default

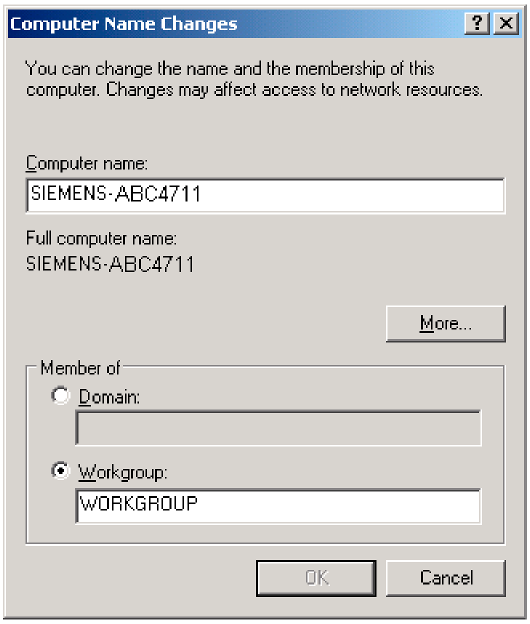
The PCU is supplied with an automatically generated computer name.

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:



Computer Name Changes

You can change the name and the membership of this computer. Changes may affect access to network resources.

Computer name: SIEMENS-ABC4711

Full computer name: SIEMENS-ABC4711

Member of

○ Domain:

● Workgroup: WORKGROUP

Figure 2-1 Changing the name of the PCU

2.1.2 How to set the IP address of the PCU 50.3

Default

NOTICE

The IP address 192.168.214.241 is set as a factory default for every PCU 50.3 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".

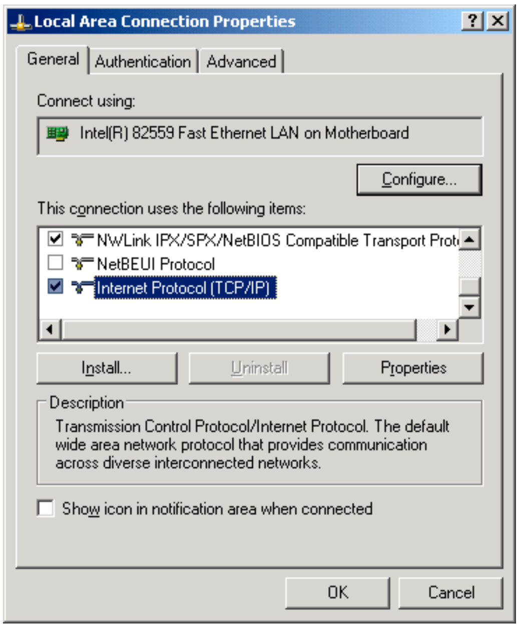
   The "Network Connections" window opens.

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:



Connect using:

Intel(R) 82559 Fast Ethernet LAN on Motherboard

This connection uses the following items:

☑ NWLink IPX/SPX/NetBIOS Compatible Transport Prot

☐ NetBEUI Protocol

☑ Internet Protocol (TCP/IP)

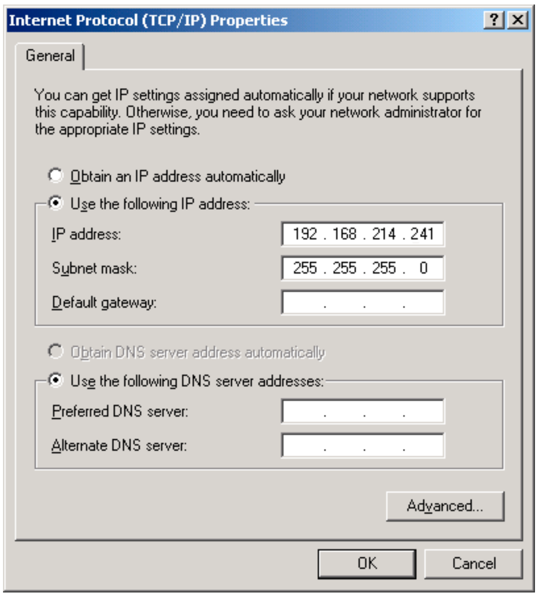
Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

☐ Show icon in notification area when connected

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:



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address:	192 . 168 . 214 . 241
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	. . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server:	. . .
Alternate DNS server:	. . .

5. Enter the required new IP address and confirm the settings with "OK".

2.1.3 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

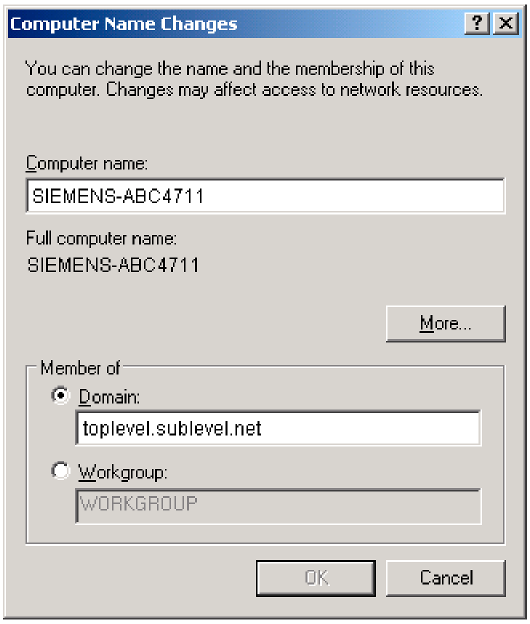
To add the PCU to a domain:

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:



Computer Name Changes

You can change the name and the membership of this computer. Changes may affect access to network resources.

Computer name: SIEMENS-ABC4711

Full computer name: SIEMENS-ABC4711

Member of

● Domain: toplevel.sublevel.net

○ Workgroup: WORKGROUP

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.

2.1.4 How to connect an external monitor

Conditions

To connect an external monitor, the following conditions 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) ... Graphics Controller"
6. Click the "Graphics Properties" button. On the "Devices" tab the external monitor corresponds to the "Monitor" selection; the OP/TP on the PCU corresponds to the "Notebook" selection.
7. Select a "Primary Device" and a "Secondary Device".
8. Close the dialog and click OK to confirm all the dialogs: The external monitor is now ready.

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.

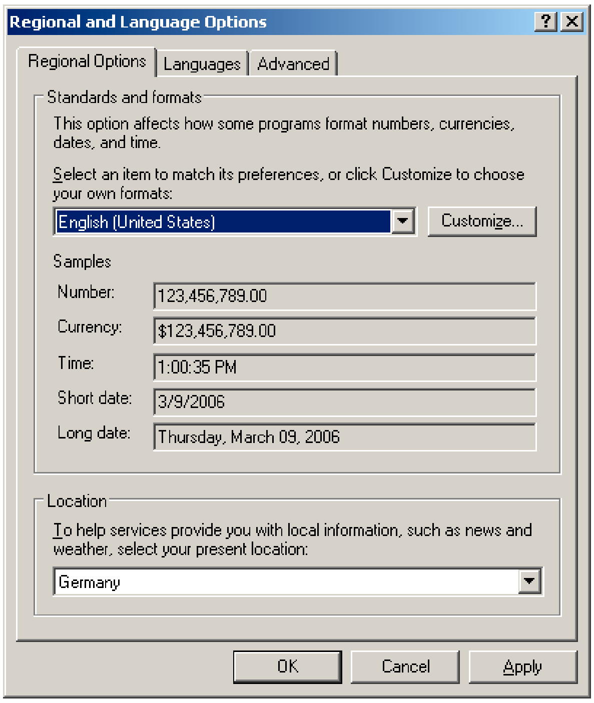
Requirement

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, dialogue boxes and keyboard layouts for the Windows system in different languages:

Select a language

After installing a language from the corresponding CD, proceed as follows:

1. Choose "Start" → "Control Panel" → "Language and Regional Options", to open the following dialogue box:



Standards and formats

This option affects how some programs format numbers, currencies, dates, and time.

Select an item to match its preferences, or click Customize to choose your own formats:

English (United States)

Samples

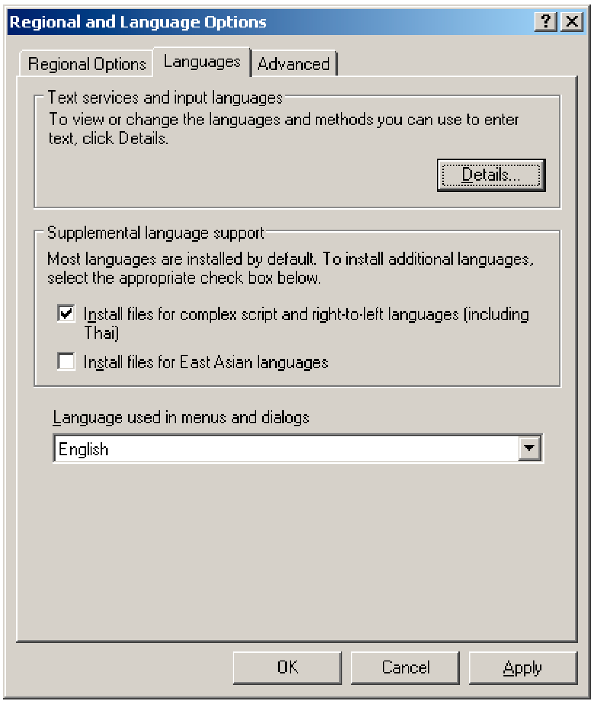
Number: 123,456,789.00
Currency: $123,456,789.00
Time: 1:00:35 PM
Short date: 3/9/2006
Long date: Thursday, March 09, 2006

Location

To help services provide you with local information, such as news and weather, select your present location:

Germany

2. Choose the "Languages" tab, in order to switch the language for the Windows XP operator interface. Under "Language used in menus and dialogues," choose the new language and confirm with OK.



Text services and input languages

To view or change the languages and methods you can use to enter text, click Details.

Supplemental language support

Most languages are installed by default. To install additional languages, select the appropriate check box below.

☑ Install files for complex script and right-to-left languages (including Thai)

☐ Install files for East Asian languages

Language used in menus and dialogs: English

3. On the "Advanced" tab, choose 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.

NOTICE
The settings for the keyboard layout and the formats for date, time and number displays on the "Regional Options" tab **must not be changed** . These settings are automatically adjusted depending on the language selected for the operator interface under HMI Advanced.

See also

How to install additional languages under Windows XP (DVD) (Page 59)

2.2.2 Storage location of HMI boot screen

Default setting

The Siemens boot screens are stored under the path below on the appropriate PCU, e.g., for the SINUMERIK 840D sl controller:

`F:\hmi_adv\ib\DATA\10700\0\<Resolution>\10700_0.bmp`

A directory tree can also be created for manufacturer-specific boot screens. The manufacturer's screens can then be saved in accordance with the schematic below:

Directory tree of manufacturer-specific boot screens:

`F:\oem\ib\DATA\<NckType>\<Resolution>\<Name>.bmp`

`<NckType>`	stands for:
0	840D
10700	840D sl
15000	840Di sl

`<Resolution>`: 640, 800, 1024 or 1280 dpi

If you want the same screen to be used each time (NCU-independent screen), it can be saved in the "default" directory in the required resolutions. If you want to be able to use different screens for different NCUs, they should be saved in the `<NckType>\<Resolution>` subdirectories using the appropriate resolution.

Screen name and resolution

`<Name>.bmp:` The name can be selected freely; only one file is permitted per directory. The screens must be created with a graphics tool in the resolution indicated by the subdirectory name and stored in the corresponding directory. The HMI software selects the screen, depending on the NCK type and the resolution of the available operator panel.

2.2.3 Displaying a customized boot screen

Directories

Set up a directory containing several boot screens (for different resolutions). This directory should be segmented into subdirectories 640, 800, 1024 and 1280, which each contain a boot screen of the appropriate resolution.

A directory containing boot screens can also be set up, likewise divided into subdirectories 640, 800, 1024 and 1280. The boot screen stored there for a particular resolution is displayed if no boot screen (including one with a lower resolution) is found in the booting-screen directory described above.

If no boot screen (including one with a lower resolution) is found, a general boot screen is displayed, which is part of the PCU-Basesoftware.

User-specific settings

The directories are customized in ServiceCenter Users under:

- Startup: Bitmaps folder
- Startup: Default bitmap folder

2.2.4 Changing the background of the service desktop

Overview

A background pattern for the service desktop is not set via the "Control Panel" (system control), but in the registry:

- Background pattern:

Key:	HKLM\SOFTWARE\Siemens\SINUMERIK\Basic software\ <version>\HMI desktop
Value:	Pattern (STRING)
Date:	<bit code>" (e.g., "0 80 114 32 0 5 39 2", see HKCU\Control Panel\Patterns) "(None)" (= NO background pattern)
Init data:	Background pattern previously set via the Control Panel
Default data:	"(None)" (if entry is not available/readable)

- Background image:

  A background screen for the service desktop is not user-specifically set via the "Control Panel" (system control), but in the ServiceCenter Users under "Service: Bitmap Folders" or "Service: Default Bitmap Folders".

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 boot screen display.

2.3 Customized settings during boot up

Overview

If an HMI program, e.g. HMI Advanced, is to be started following boot up, the settings and functionality differ from those that apply if the service desktop is to be started.

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 can be configured in file E:\Windows\System.ini.

Filtering keystroke sequences:

Section:	MMC103Keyb
Key:	SeqAct
Value:	<bit mask> (= 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> (= function keys to be filtered, specified in accordance with the comment in E:\Windows\System.ini)
Init value:	255

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 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 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 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 PCU 50.3 with SITOP UPS module

Conditions

- PCU-Basesoftware WinXP V08.00.00 or higher for USB port
- HMI Advanced of V07.01.00 or higher
- SITOP software, version 2.5.2.4or higher is installed:

  SITOP software is available to download from: www.automation.siemens.com/sitop.
- SITOP monitor/configuration program is installed:

  To enable this, the SITOP software must be copied to the E:\SITOP directory set up on the PCU. This directory already contains PCU tools required for shutdown on the SITOP UPS. If this directory does not exist in an older version of the PCU-Basesoftware, it must be created so that it will be compatible for any subsequent updating of the PCU-Basesoftware.
- UPS USB driver for Windows XP is installed:

  Installation is described in the relevant SITOP documentation. The documentation is part of the SITOP software download package.
- The SITOP UPS hardware is connected.

NOTICE

The SITOP software version 3.1.0.6 can be operated with PCU-Basesoftware V08.02.00.01 or higher subject to the following conditions:

- SITOP software must not be installed as a Windows service, but must be started as a normal application (as described in the chapter titled "Starting and configuring the SITOP monitor").
- The SITOP service prevents the correct switch-off procedure being performed on the SITOP UPS module when PCU-Basesoftware is installed.

Application

If the supply voltage at the PCU dips, the SITOP UPS modules below could maintain operation for a limited period if a backup battery is being used, allowing the PCU to be properly shut down before the battery is exhausted:

Name	Order number (MLFB)
SITOP POWER DC UPS module 15 (USB interface)	6EP1931-2EC41

Note

Information about the test environment for machine OEMs

The "SITOP UPS" function has been tested in the standard configuration of HMI Advanced. When installing add-on or OEM software components, the shutdown procedure of the complete system has to be checked by the operator.

2.4.1 Starting and configuring the SITOP monitor

Windows boot up

The SITOP monitor has to be started by Windows automatically during booting. A new value has to be entered for the SITOP monitor under the key below in the Windows registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
          ValueName: SITOP
          ValueType: REG_SZ
          Value Data: E:\SITOP\SITOP_DC_USV.exe
```

A script file, sitop.reg, is located in the E:\SITOP directory. The required key is entered into the registry automatically if this file is executed.

The SITOP monitor is started automatically once the PCU is restarted. The next installation step is to configure the monitor.

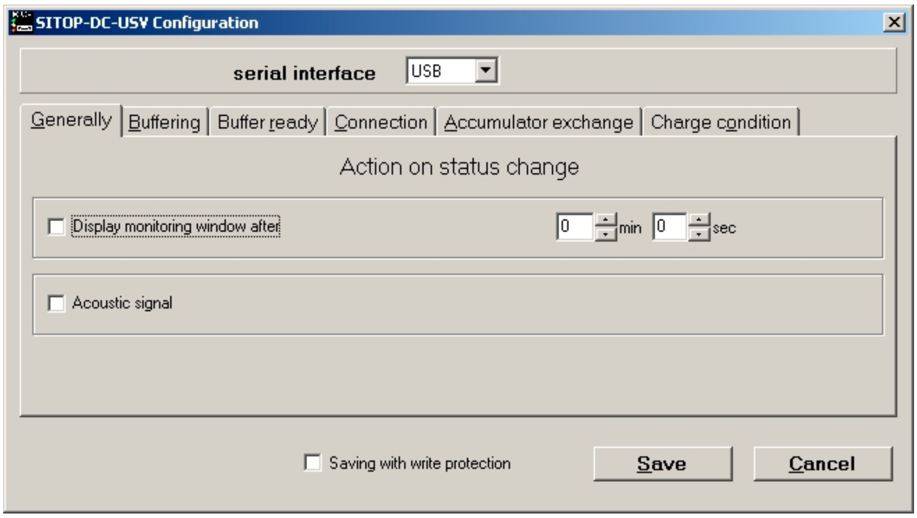
NOTICE
The SITOP monitor must not be started via the Windows Autostart directory.

General settings

The following settings must be made in the SITOP monitor configuration dialogue box:

- Parameterization of the interface: For module with USB port
- Parameterization of the change action:

  The monitoring window display must be deselected, as this function can lead to sporadic faults on the HMI operator interface.



serial interface: USB

□ Display monitoring window after 0 min 0 sec

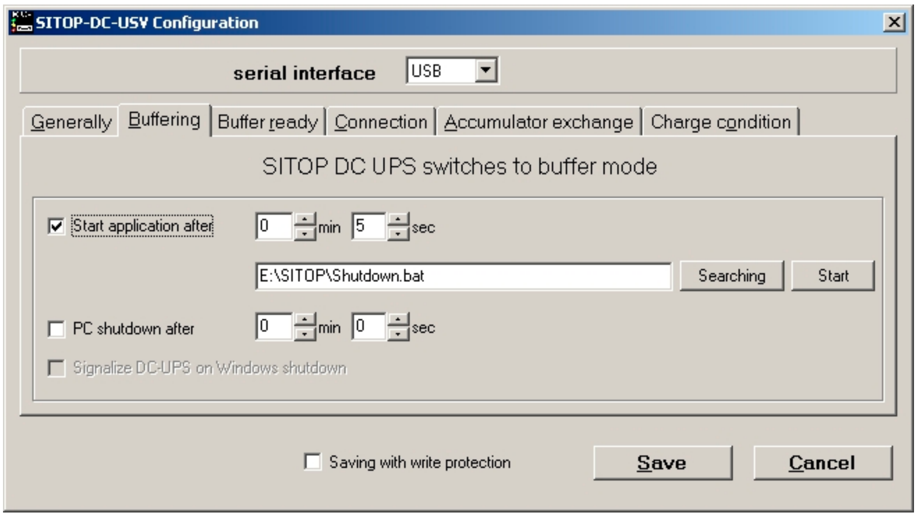
□ Acoustic signal

□ Saving with write protection

Figure 2-3 Configuration of the SITOP monitor: General settings

Parameterization of the SITOP monitor

Enter the path of the program that ensures that HMI Advanced and the PCU shut down properly in the event of a power failure into the parameter area of the buffer.
E:\SITOP\Shutdown.bat



SITOP-DC-USV Configuration

serial interface: USB

SITOP DC UPS switches to buffer mode

☑ Start application after 0 min 5 sec

E:\SITOP\Shutdown.bat

□ PC shutdown after 0 min 0 sec

□ Signalize DC-UPS on Windows shutdown

□ Saving with write protection

Figure 2-4 Configuration of the SITOP monitor: Buffering settings

Alternatively, the following setting can be used: E:\Windows\system32\hmiexit.exe

2.4.2 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 (PCU basic software XP 08.00.00 or higher)

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 +22V (fixed)
2		○	+1V	
3	○		+0.5V	
4		○	+1V	End-of-charge voltage + 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
3		○	+160 s	
4		○	+80s	
5		○	+40 s	
6		○	+20 s	
7		○	+10 s	+5 s fixed
8	●			Disconnection
9		○		Accumulator operating state on/off

Legend: ○ Delivery condition setting
● Setting for operation on the PCU 50

"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 +22V (fixed)
2		○	+1V	
3	○		+0.5V	
4		○	+1V	° ° End-of-charge voltage + 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 +5 s fixed
3	●		+160 s	
4		○	+80s	
5		○	+40 s	
6		○	+20 s	
7		○	+10 s	
8	●			Disconnection
9		○		Accumulator operating state on/off

Legend:	○	Delivery condition setting
	●	Setting for operation on the PCU 50

2.4.3 Configuration for exiting the HMI

HMI monitoring

Exiting of the HMI advanced is monitored by a separate application, hmiexit.exe. This application is started implicitly via the shutdown.bat batch file. In case of error, the application forces the operating system to shut down. An error occurs if the HMI cannot be exited within the configured delay.

Optionally, the parameters below can be set for hmiexit in file:

```
E:\SITOP\hmiexit.ini.
[Actions]
#Waiting time in seconds for closing HMI Advanced applications
Wait = 120
# Action on expiration of the waiting time
ForceShutdown = True
```

These default settings only need to be changed if it takes longer than 120 seconds to exit the HMI applications in an OEM installation. This configuration is not usually changed.

Configuring the "EXIT" softkey

During production, the option to shut down the HMI via the "EXIT" softkey in the operating area menu should be disabled, as this function cannot be synchronized with the UPS module.

The "EXIT" softkey is disabled by entering ExitButton=False in the regie.ini file.

Hibernate

The operating system’s hibernate mode is suspended when operating the UPS, as the USB interface always has to be active for the UPS module.

Note

More information can be found in the product descriptions with the corresponding order number.

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.

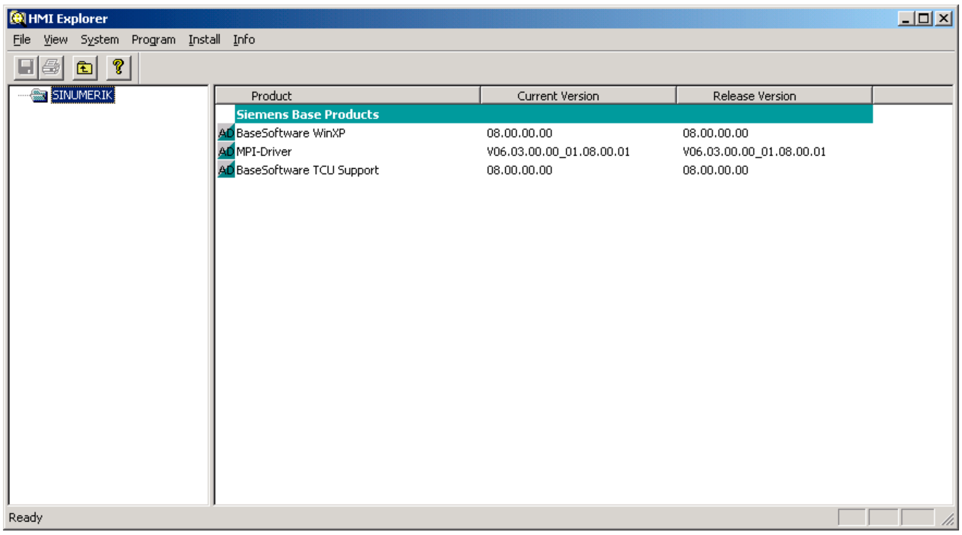
The installation directories are evaluated in the sequence described in the parameters file. If the parameters file is missing or it does not contain a [SetupDirs] section, the preset "D:\Install" installation directory is considered instead.

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 from HMI Explorer.



Product	Current Version	Release Version
Siemens Base Products		
BaseSoftware WinXP	08.00.00.00	08.00.00.00
MPI-Driver	V06.03.00.00_01.08.00.01	V06.03.00.00_01.08.00.01
BaseSoftware TCU Support	08.00.00.00	08.00.00.00

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.

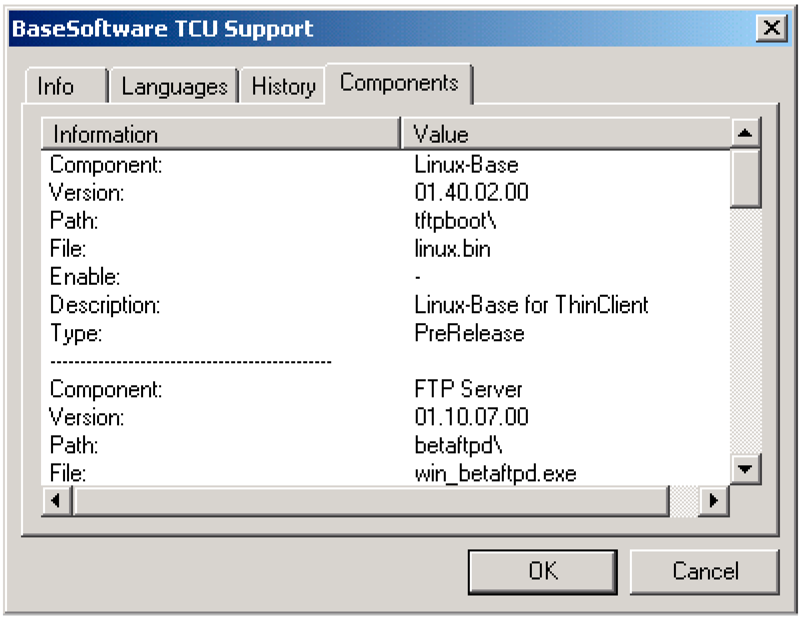
- History of the product

  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



Information	Value
Component:	Linux-Base
Version:	01.40.02.00
Path:	tftpboot\
File:	linux.bin
Enable:	-
Description:	Linux-Base for ThinClient
Type:	PreRelease

Component:	FTP Server
Version:	01.10.07.00
Path:	betaftpd\
File:	win_betaftpd.exe

Figure 3-2 "Components" dialogue box (example)

3.3 How to install and authorize SIMATIC STEP 7

Overview

SIMATIC STEP 7 V5.4 SP1 can also be installed on the PCU.

Delivery item:	SIMATIC STEP 7 V5.4 SP1
Components:	SIMATIC STEP 7 V5.4 SP1 and Add-on for SINUMERIK 840Di sl/840D sl/840D
Type of delivery:	1 installation DVD
System requirements:	PCU-Basesoftware V8.0 or higher A network connection or a connection to a DVD drive is required. **Recommendation:** Mouse port

Note

The "MPI driver" package available on the PCU is part of the HMI software and must not be uninstalled!

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 now be started from the HMI Advanced operator interface (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 SP2 to:

- Subsequently install Windows components
- Re-establish the delivery condition of the PCU without HMI Advanced
- 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 SP2 Windows XP ProEmbSys operating system, incl. SP2 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 • Ghost image of the delivery condition of the PCU basic software Windows XP for PCU 50.3 without HMI Advanced and other application software. • Ghost image for creating an "Emergency Boot System" (identical to the directory D:\EBOOT on the PCU)
3_MUI_1	Chinese (simplified) Traditional Chinese Japanese Korean Romanian Slovakian
4_MUI_2	Danish German French Dutch Italian Spanish Swedish
5_MUI_3	Brazilian Portuguese Finnish Polish Russian Czech Turkish Hungarian
EULA TERMS Certificate of authenticity	Contained 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 of the HMI Advanced operator interface is independent of this. It is set independently of this under "Start-up" → "HMI" → "Change Language".

Backing up and restoring data 4

4.1 Backing up and restoring data

Overview

The entire contents of hard disks can be saved as a disk image using the Symantec Ghost utility. These disk images can be stored on various storage media and the data restored to the hard disk at a later date.

PCU replacement hard disks and complete PCU hard disks are supplied by the plant with Symantec Ghost already installed. Symantec Ghost is used for the data storing and restoring processes described in the following sections.

More information is available on the Internet at: http://www.ghost.com/

4.1.1 How to create a service system for PCU

Use

In case servicing is needed, create a portable service system as an "Emergency Boot System" (EBS) on a USB memory store on the basis of WinPE.

Recommendation:

It is better to use SIMATIC PC USB-FlashDrive.

Creating a service system

The Ghost image is available on the hard disk under D:\Eboot in order to create the service system for a PCU 50.3 on a USB memory store.

Use the following procedure:

1. Start the PCU 50.3in the service mode.
2. Plug a USB memory store with at least 256 MB into one of the four USB ports of the PCU 50.3.
3. Launch Ghost32.exe in directory E:\Tools
4. Select Ghost: Local → Disk → From Image. As the source, select D:\Eboot\eboot.gho and as the destination select the USB memory store (recognizable by its storage capacity).

   Result:

   After successfully transferring eboot.gho to the USB memory store, the service system for the PCU is ready to use.

Using the same procedure, create a service system on a PG/PC. For this, the Symantec Ghost program must be installed on the PG/PC.

Booting up the service system

1. Plug the EBS into one of the rear USB ports of the PCU 50.3.
2. Press the <ALARM CANCEL> key on the operator panel or the <ESC> key on an external keyboard while the BIOS of the PCU is booting up in order to display the "Boot Menu".
3. Select the entry "USB-HDD: XXX " from the list of available media.

   Result:

   The PCU boots up from the service system and the ServiceCenter starts.
4. First select "Start" to start the ServiceCenter and then "Backup/Restore Disk Image", to restore the hard disk using the disk image.

NOTICE

During booting:

- The boot-up of the PCU from the EBS via the front USB interface (=USB V1.1; rear USB V2.0) of a directly connected OP is also possible but it is considerably slower.
- It is not possible to boot up the PCU from the EBS via the USB interface of a TCU.
- It is not possible to save network settings on the EBS.
- The EBS is not capable of functioning if a DVI monitor is connected directly to the DVI interface of the PCU 50.3. VGA mode is only possible via a DVI → VGA adapter.

Booting up the service system in "Headless"mode

A PCU 50.3 with BIOS version 05.01.11 is required.

The PCU 50.3 is installed in the control cabinet and is only operated with a TCU that has no OP/TP:

- After locking the hard disk, you can boot up from the USB service system and power up using WinPE.
- Then release the hard disk again and proceed in the manner described above.

See also

How to backup and restore the hard disk (Page 67)

4.1.2 Starting ServiceCenter Backup Restore

Starting ServiceCenter Backup Restore

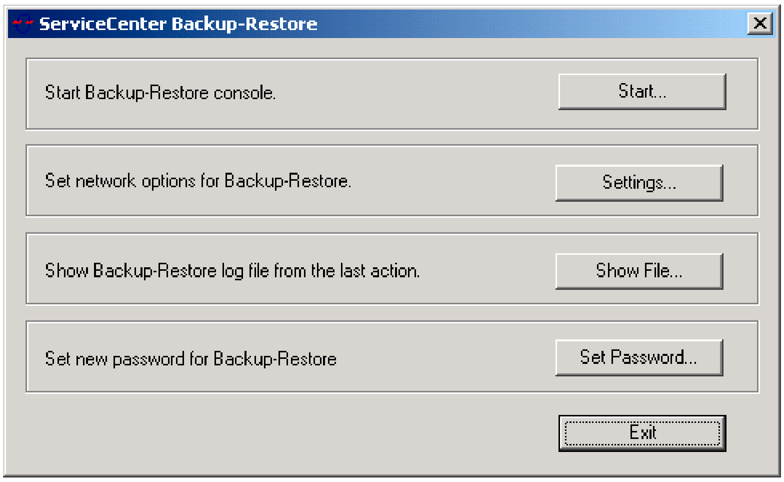
You start the ServiceCenter Backup Restore for the following tasks:

- Backing up/restoring data
  - Starting via linking from the service desktop
  - Starting during boot up via entry in the "background" under SINUMERIK
- From the service system in the event of service
- When installing a replacement hard disk

In order to ensure that the user is authorized to carry out service tasks, direct access is password-protected during boot up of the PCU. This password can be changed.

NOTICE
If you start the ServiceCenter Backup Restore via the service desktop from a service system or when installing a replacement hard disk, no password is needed.

The following dialog appears after you double-click the ServiceCenter Backup Restore link on the service desktop:



ServiceCenter Backup-Restore

Start Backup-Restore console.

Set network options for Backup-Restore.

Show Backup-Restore log file from the last action.

Set new password for Backup-Restore

Start ... With "Start," you start the shutdown of the system and the start of the ServiceCenter.

Settings ... With "Settings," you open the dialogue box for network settings.

Show File ... Under "Show File," you can view the log of the last data back up.

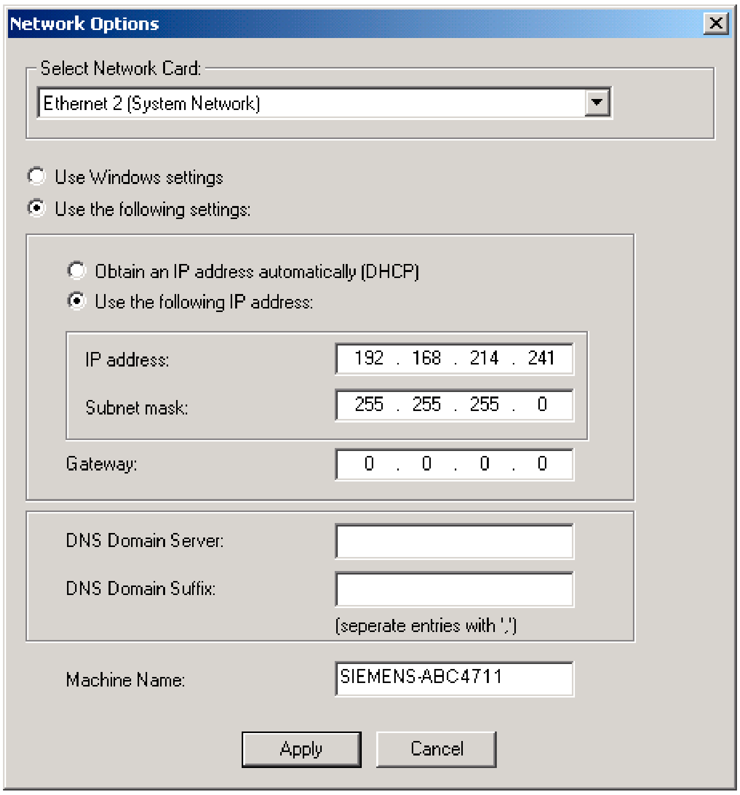
Set password ... This is where you enter a new password for ServiceCenter Backup Restore.
(the default is the same as for the service user "auduser".)

Exit Cancel and return to the service desktop.

Figure 4-1 Starting service

Network settings

1. To connect the PCU to a programming device or PC, choose "Settings" in order to check or reset the set IP addresses.
2. Select "Use Windows settings", to keep the factory defaults. (This is the default here as well.)



Network Options

Select Network Card: Ethernet 2 (System Network)

○ Use Windows settings
● Use the following settings:

○ Obtain an IP address automatically (DHCP)
● Use the following IP address:

IP address: 192 . 168 . 214 . 241
Subnet mask: 255 . 255 . 255 . 0
Gateway: 0 . 0 . 0 . 0

DNS Domain Server:
DNS Domain Suffix:
(seperate entries with ',')

Machine Name: SIEMENS-ABC4711

Figure 4-2 Network settings

3. Select "Use the following settings", to set a new configuration:
   - With "Obtain an IP address automatically (DHCP)," you receive an automatically assigned IP address from your DHCP server.
   - With "Use the following IP address," you enter an IP address in the range of 192.168.214.250 – 254 using subnet screen form 255.255.255.0.
4. To activate a DNS name service, specify the server's IP address under "DNS Domain Server" and the extension, e.g. "network.com" under "DNS Domain Suffix".

   The default is ".local" if you don't enter anything else.

Note

Changes to the network settings that you make here only become effective after you reboot the PCU.

On the other hand, if you make changes to the network settings from service desktop, they are immediately adopted.

See also

System features (Page 7)

4.1.3 Select service task

Selecting the service task

After start-up of the ServiceCenter, the following dialog opens:

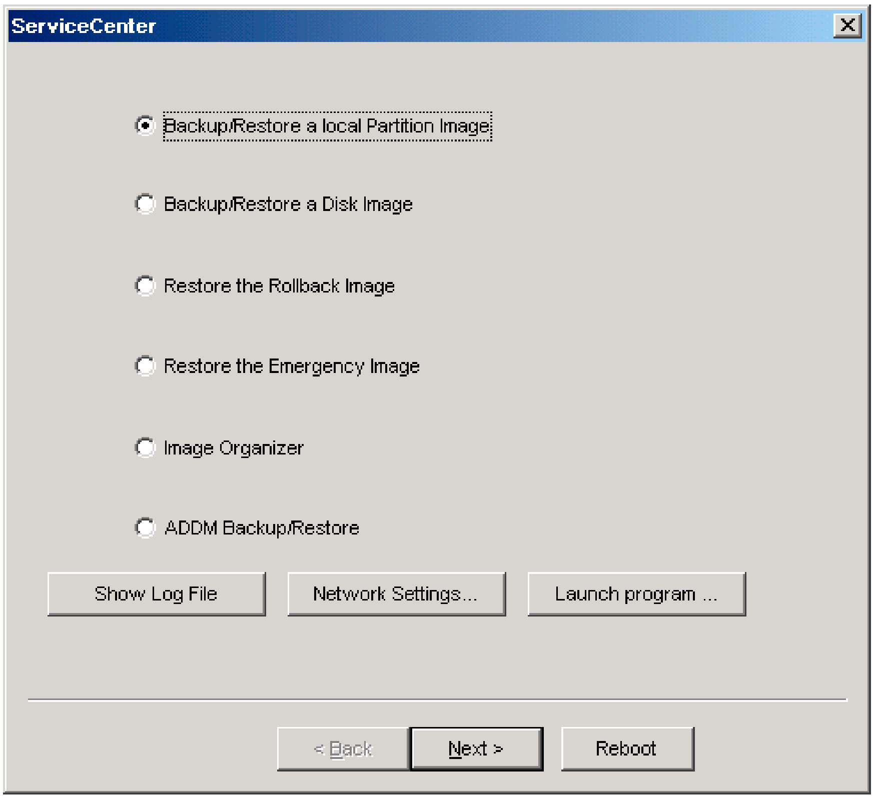

ServiceCenter
● Backup/Restore a local Partition Image
○ Backup/Restore a Disk Image
○ Restore the Rollback Image
○ Restore the Emergency Image
○ Image Organizer
○ ADDM Backup/Restore
Show Log File
Network Settings...
Launch program ...
< Back
Next >
Reboot

Figure 4-3 ServiceCenter Selection

Select from the following service tasks:

- Backup/Restore a local Partition Image
- Backup/Restore a Disk Image
- Restore the Rollback Image
- Restore the Emergency Image
- Image Organizer

Show log file

This option opens file bacres.txt, which contains a log of all backup records.

Network Settings

With "Network Settings," you open the dialogue box for network settings.

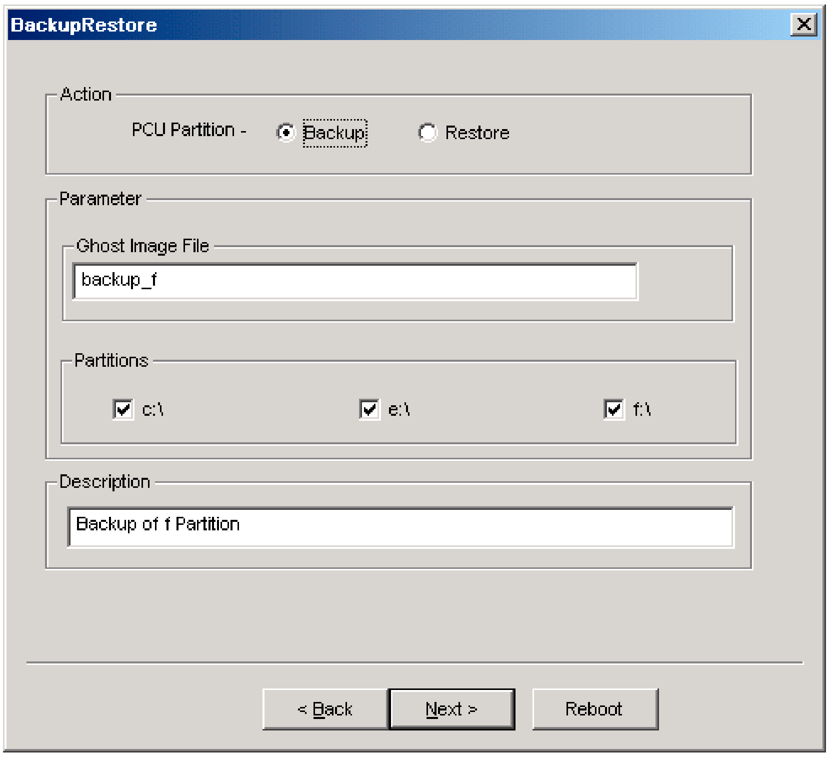
Launch Program

To start a program in service mode, enter the program name here, e.g. "cmd" for starting a DOS shell.

4.1.4 How to backup and restore local partitions

Backing up partitions

1. Select the "Backup" action from "Backup/Restore a local Partition Image" to backup an image of one or more C, E, and F partitions locally on the D:\Images partition of the hard disk:



BackupRestore

Action: PCU Partition - ● Backup ○ Restore

Parameter

Ghost Image File: backup_f

Partitions: ☑ c:\ ☑ e:\ ☑ f:\

Description: Backup of f Partition

Figure 4-4 Local Partition Backup

2. Select the partitions for which an image is to be produced.
3. Before the backup is started, the size of each partition will be displayed in the next dialogue.

   Recommendation:

   If you wish to save the backup file and restore it later, we recommend that you always create a complete image of partitions (C, E and F).

Restoring partitions

Select the "Restore" action from "Backup/Restore a local Partition Image" to restore an image of one or more C, E, and F partitions locally from the D:\Images partition:

Restore the Rollback Image

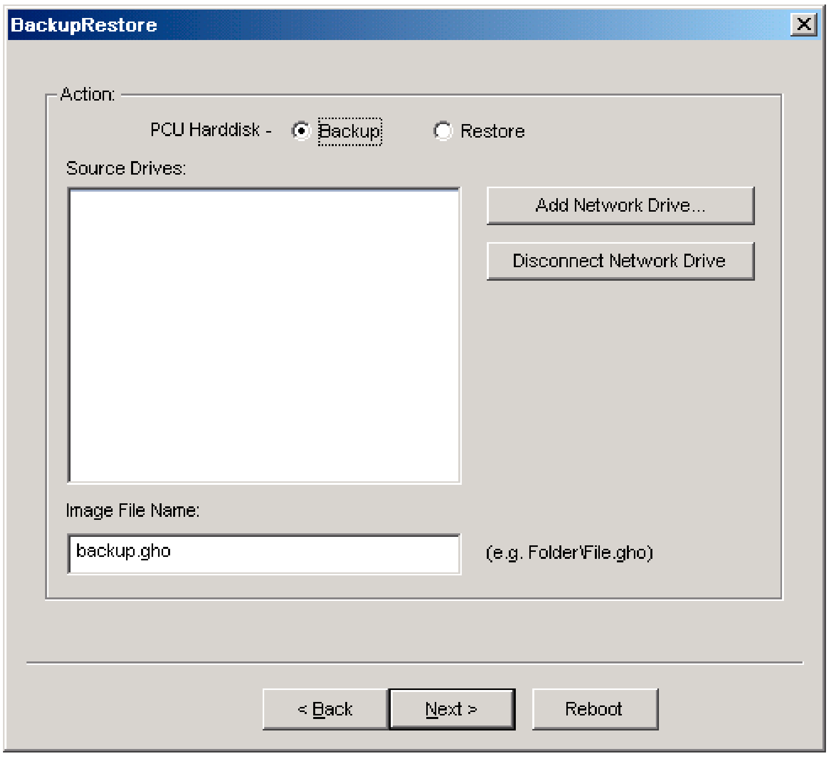
To restore the most recently saved image, namely the current image ("Rollback Image"), select "Restore the Rollback Image".

The "Rollback Image" is the last created back-up of a partition.

4.1.5 How to backup and restore the hard disk

Backing up the hard disk

Select "Backup/Restore a Disk Image" to backup an image of the hard disk using the network connection:



BackupRestore

Action:

PCU Harddisk - ● Backup ○ Restore

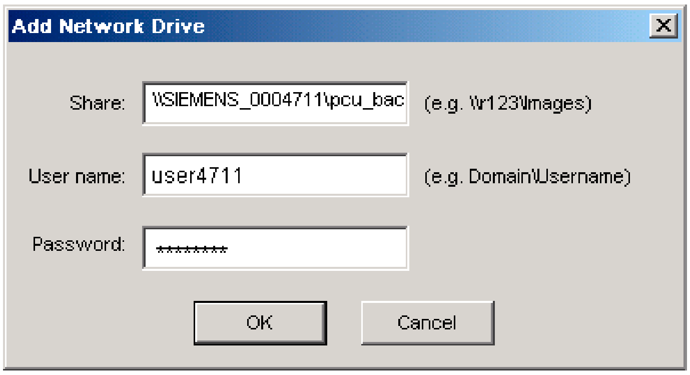
Source Drives:

Image File Name: backup.gho (e.g. Folder\File.gho)

Figure 4-5 Backup hard disk via the network

1. In order to establish a network connection with access to a released drive, select "Add Network Drive" and specify the name of the file for "Image File Name."
2. Under "Share," enter the computer name and the released directory.

3. To receive access rights, enter a user name and password.



Share: \\SIEMENS_0004711\pcu_bac (e.g. \\r123\Images)

User name: user4711 (e.g. Domain\Username)

Password: ********

Figure 4-6 Drive Connection

4. Under "Options," select whether the disk image that is to be created is divided into several files of a certain size, so that these files can fit on one CD.

Restoring a hard disk

Select the "Restore" action from "Backup/Restore Disc Image" to restore an image. Click "Next>" to be prompted:

See also

Requirements (Page 71)

4.1.6 Restoring system data from "Emergency Image"

Restoring system data

Select "Restore the Emergency Image", to restore the emergency image. This image must contain the back-up of partition E and can also contain a back-up of partitions C, D or F. It is provided in the event that only the system on partition E: is defective. The user data on partition F: are kept in the current status.

⚠CAUTION

The reading in of an "Emergency Image" from partition E: can only take place if no additional software has been installed or configured after this back up is created or the registry entries of all the applications that are on partition F: must be included in the image.

Use the "Image Organizer" function only to identify an image as an emergency image that fulfills these conditions.

To restore the system data with "Emergency Image", the PCU must be booted from the service system (EBS).

Manage images

Select “Image Organizer” to mark out one image contained in the displayed list as the emergency image, or to delete an existing image.

See also

How to create a service system for PCU (Page 61)

4.2 Saving the HMI Advanced environment

Setting the original SINUMERIK HMI environment

The "Original SINUMERIK HMI Environ" function is available as a script file on the service desktop. The original delivery condition is set up, i.e., the contents of the directories below are saved, when this function is executed:

- C:\RUNOEM
- F:\ADD_ON
- F:\OEM
- F:\USER

Then the directories are cleared.

Setting the current SINUMERIK HMI environment

The "Current SINUMERIK HMI Environ" function is available as a script file on the service desktop. When this function is executed, restoration of the original settings is canceled, i.e., the saved directory contents are copied back.

4.3 Operating the service PC or the PG on the network

Applications

For the following applications, you will, for example, need a connection in the system network between the PCU and a PG/PC:

- To store a backup image from the hard disk of a PCU 50 on a PG/PC.
- To restore a PCU 50 hard disk via the CD-ROM drive of a PG/PC.
- To commission a replacement hard disk.

4.3.1 Requirements

Overview

The following figures show the typical connection options in the system network:

- PCU to "Eth 2" with service PG/PC, directly, using a crossed Ethernet cable
- PCU to "Eth 2" with service PG/PC, via a switch, using an un-crossed Ethernet cable

If you want to connect the service PG/PC via a company network (Eth 1), contact your network service center.

Meaning of the connections:

○	Eth 1 as a DHCP client
●	Eth 2 as a DHCP server
■	Eth 2 with a fixed IP address
Green connection	Uncrossed Ethernet cable
Gray connection	Crossed Ethernet cable (crossover)

Configuration with PG/PC directly to PCU

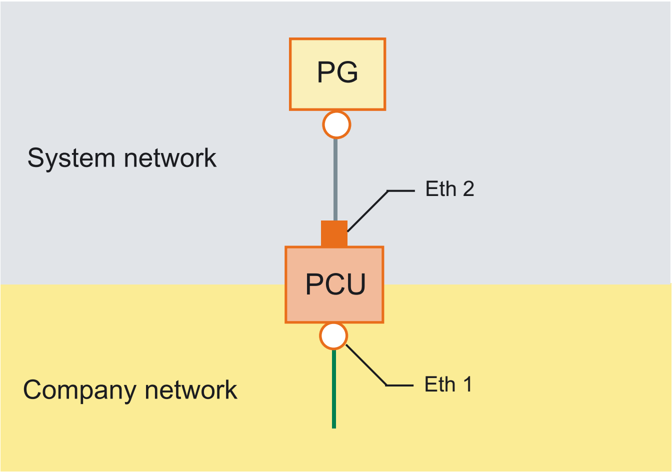


Figure 4-7 Connecting a PG directly to a PCU

Configuration with PG/PC and switch to PCU

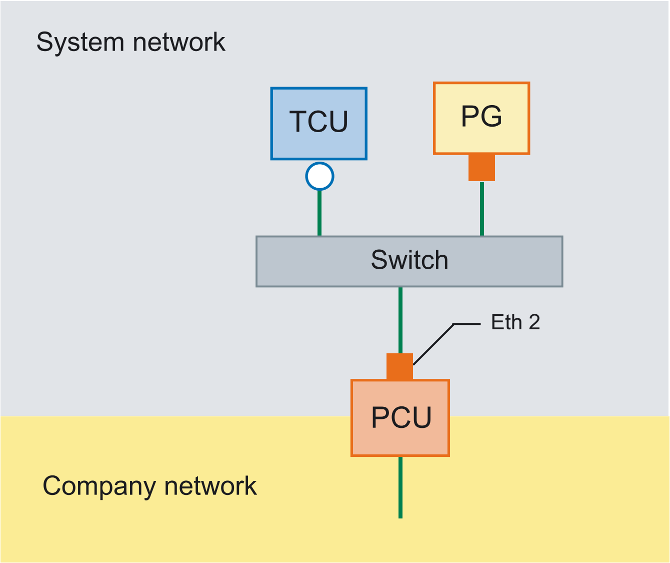


Figure 4-8 Connecting a PG via a switch to a PCU

Note

If a PCU is switched off and on again without its own OP/TP including TCU, and if the PCU boot-up is supposed to take place from the service system (EBS), an external VGA monitor and keyboard are needed in order 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 XP:

- Connecting a PG/PC to a PCU 50.3 as per one of the configurations in the figures above.
- The network protocol used is: TCP/IP.
  TCP/IP is already pre-configured in the basic PCU software.
- Setting up IP addresses on the same subnetwork.
- Releasing a directory on the PG/PC for network access.

On the PCU under WinPE:

- Start the ServiceCenter under WinPE on the PCU 50.3.
- Establish a network connection to the released directory of the PG/PC.
- Using the "Backup" function, a ghost image of the PCU hard disk is saved in the released directory of the PG/PC in the event of a need for service.
- With the "Restore" function, the hard disk of the PCU 50.3 is restored from a ghost image in the released directory of the PG/PC.

See also

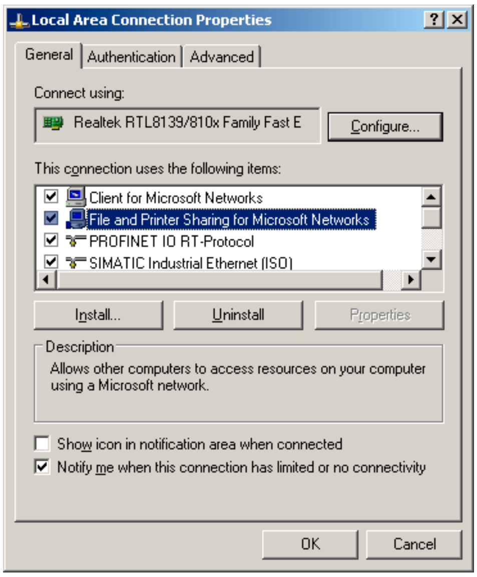
How to backup and restore the hard disk (Page 67)

4.3.2 How to connect a PC/PG to a PCU within the system network

Settings on a PG/PC with Windows XP

On a PG/PC, the following settings must be made:

1. Select "Control Panel" → "Network Connections" → "Local Area Connection Properties", then you will see the following dialogue box:



Local Area Connection Properties

General | Authentication | Advanced

Connect using:
Realtek RTL8139/810x Family Fast E — Configure...

This connection uses the following items:
- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ PROFINET IO RT-Protocol
- ☑ SIMATIC Industrial Ethernet (ISO)

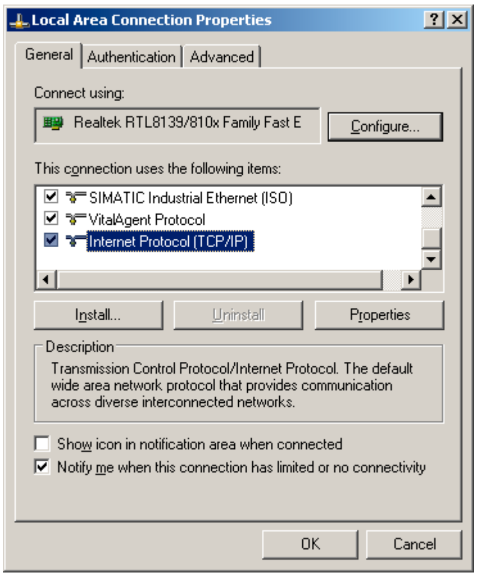
Install... | Uninstall | Properties

Description
Allows other computers to access resources on your computer using a Microsoft network.

☐ Show icon in notification area when connected
☑ Notify me when this connection has limited or no connectivity

OK | Cancel

2. Check to see whether "File and Printer Sharing ..." is selected, so that directories can be released and then select "Internet Protocol (TCP/IP)".



Local Area Connection Properties

General | Authentication | Advanced

Connect using:
Realtek RTL8139/810x Family Fast E — Configure...

This connection uses the following items:
- ☑ SIMATIC Industrial Ethernet (ISO)
- ☑ VitalAgent Protocol
- ☑ Internet Protocol (TCP/IP)

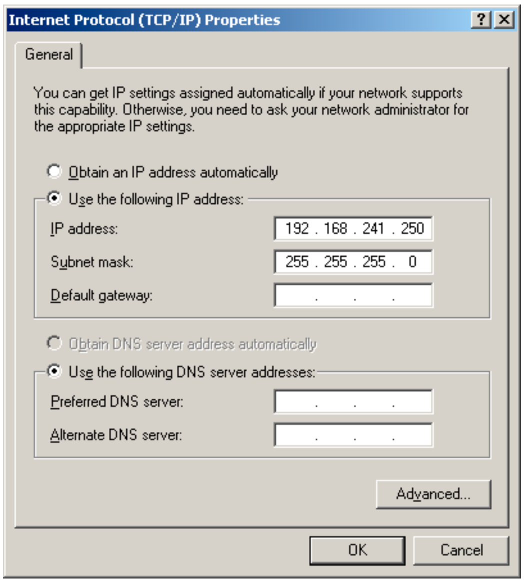
Install... | Uninstall | Properties

Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

☐ Show icon in notification area when connected
☑ Notify me when this connection has limited or no connectivity

OK | Cancel

3. Open the "Properties" dialogue box and select the option "Use the following IP address", in order to enter an IP address, such as 192.168.214.250 and the subnet screen form 255.255.255.0.



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

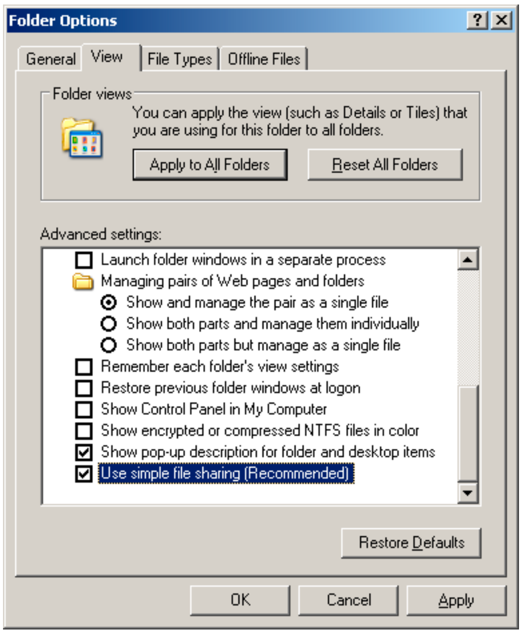
IP address: 192 . 168 . 241 . 250
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: . . .
Alternate DNS server: . . .

4. Select "Control Panel" → "System" → "Computer Name" tab to view the computer name of the PCU: e.g. SIEMENS-ABC4711

5. Select "Control Panel" → "Folder Options" → "View" and activate "Use simple file sharing (Recommended)", to avoid problems with the release of the directory.



Folder Options

Folder views

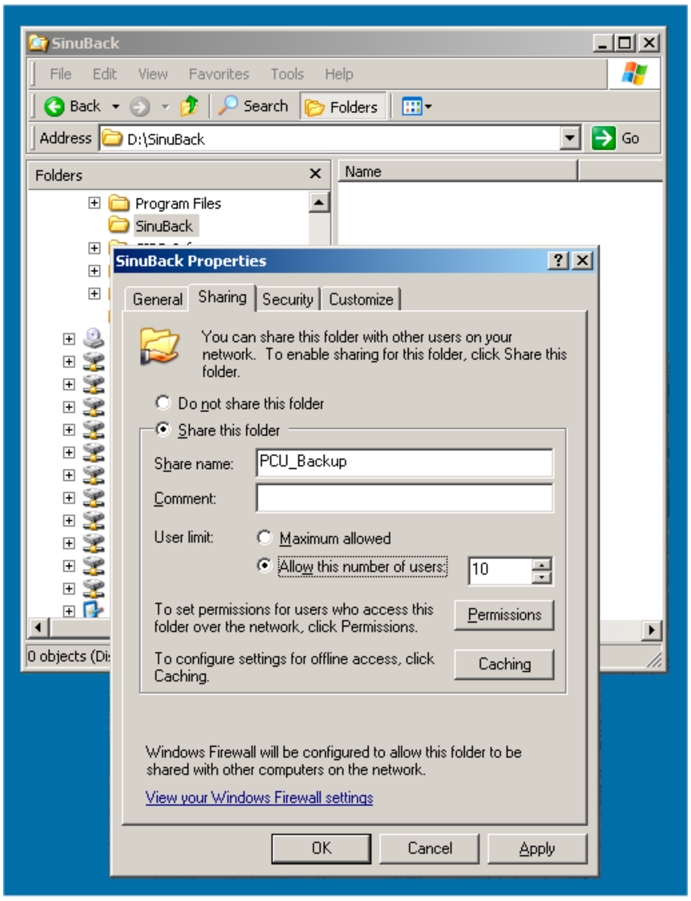
You can apply the view (such as Details or Tiles) that you are using for this folder to all folders.

Advanced settings:

- □ Launch folder windows in a separate process
- Managing pairs of Web pages and folders
  - ● Show and manage the pair as a single file
  - ○ Show both parts and manage them individually
  - ○ Show both parts but manage as a single file
- □ Remember each folder's view settings
- □ Restore previous folder windows at logon
- □ Show Control Panel in My Computer
- □ Show encrypted or compressed NTFS files in color
- ☑ Show pop-up description for folder and desktop items
- ☑ Use simple file sharing (Recommended)

Releasing directory for network access (Windows XP)

1. Create a directory on a local drive; e.g. D:\PCU_Backup
2. Using the right mouse key, open the "Properties" dialogue box of the directory and the "Sharing" tab.



You can share this folder with other users on your network. To enable sharing for this folder, click Share this folder.

○ Do not share this folder

● Share this folder

Share name: PCU_Backup

Comment:

User limit: ○ Maximum allowed

● Allow this number of users: 10

To set permissions for users who access this folder over the network, click Permissions.

To configure settings for offline access, click Caching.

Windows Firewall will be configured to allow this folder to be shared with other computers on the network.

View your Windows Firewall settings

3. Select "Share this folder". As a share name (release name), the directory name is used, e.g. PCU_Backup.

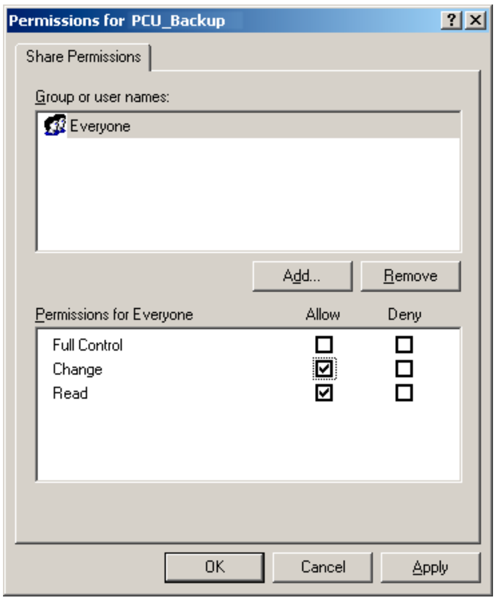
   If the directory name is changed, the new name must be specified when connecting the drive!

Note

Ensure there is sufficient free memory on the hard disk of the PG/PC to be able to save the ghost image when creating a back-up.

---

4. Select "Permissions" and activate the "Change" square for all the users in the "Allow" column so that files can be saved in this directory (e.g. the ghost image).



Everyone

Permissions for Everyone	Allow	Deny
Full Control	☐	☐
Change	☑	☐
Read	☑	☐

Procedure on the PCU 50.3

The following steps must be carried out on the PCU:

1. Start the ServiceCenter with "Start Backup/Restore console".
2. Maintain the pre-setting of the "Network Options" on the PCU:

IP address of PCU:	192.168.214.241	with subnet screen form 255.255.255.0
IP address of PG/PC:	192.168.214.250	with subnet screen form 255.255.255.0

1. In the ServiceCenter, select the service task "Backup/Restore a Disk Image".
2. Establish a network connection to the released directory, e.g. \\SIEMENS-ABC4711\PCU_Backup.
3. Restore the hard disk of the PCU using the ghost image.

Note

If the transfer is interrupted during the "Restore" process, no consistent system is available on the hard disk, i.e. the "Restore" process cannot be repeated because the PCU no longer boots up.

In this event, the "Emergency Boot System" on the USB memory store is used.

See also

How to backup and restore the hard disk (Page 67)

How to create a service system for PCU (Page 61)

4.3.3 How to connect a PC/PG to an NCU within the company network

Add route ... in WINPE

This function is required for the following application: Backing up/restoring ghost images

Example configuration

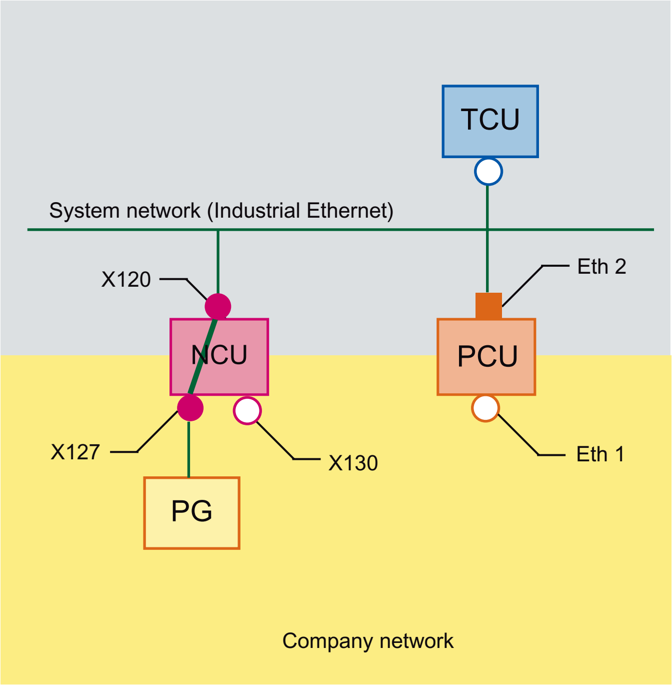


Figure 4-9 Network routing

Conditions

The following conditions must be met in order to use this function:

- The PG must be connected to X127 of an NCU in the system network.
- The NCU and PCU must be connected via the system network.
- You activate routing on the NCU via X127:

  basesys.ini (in the /card/user/system/etc directory),
  parameter EnableSysNetToIBNForwarding=1

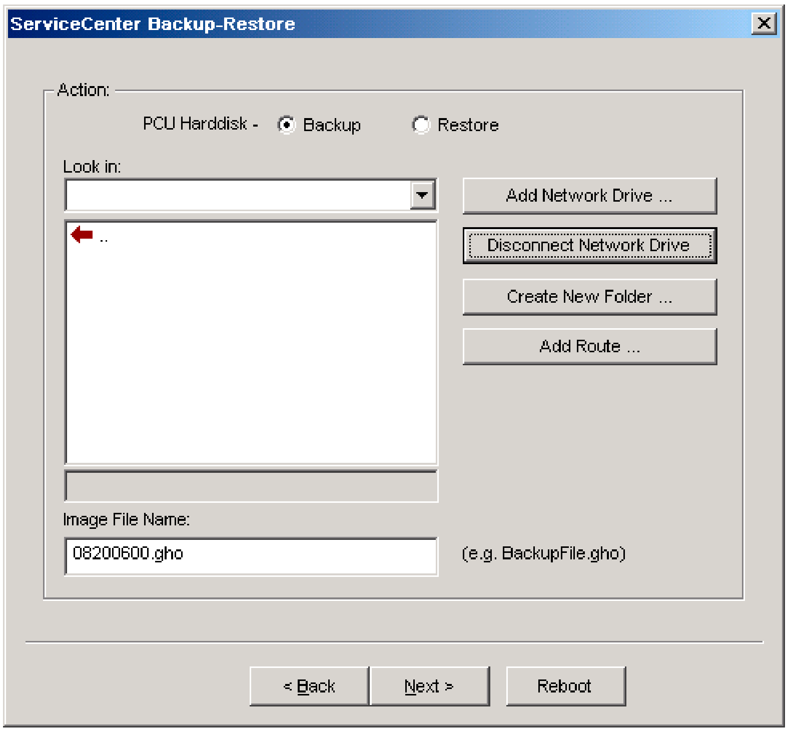
  OR

  System Network Center, "System Basics" tab,
  parameter IBN network (X127) settings: "Forwarding from system network" **enable**
- Enable a directory on the PC/PG.
- The user who logs on must be one of the recognized Windows users on the PC/PG; for example, auduser is **not** recognized on a PC.

Procedure

Activate routing prior to establishing the network connection:

1. Press the "Add Route ..." button.



ServiceCenter Backup-Restore

Action:

PCU Harddisk - ● Backup ○ Restore

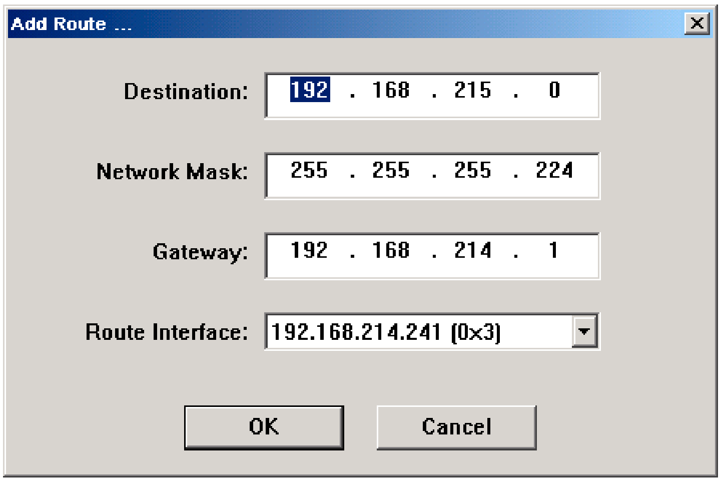
Look in:

Image File Name: 08200600.gho (e.g. BackupFile.gho)

Figure 4-10 ServiceCenter Backup-Restore

2. Establish the network connection using "Add Network Drive ...".

   In the example: \\192.168.215.2\<sharename>



Add Route ...

Destination: 192 . 168 . 215 . 0

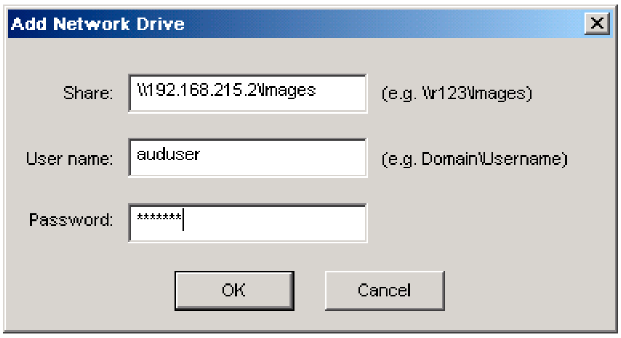
Network Mask: 255 . 255 . 255 . 224

Gateway: 192 . 168 . 214 . 1

Route Interface: 192.168.214.241 (0x3)

Figure 4-11 Dialog: Add Route ...

3. Logon and password details for a local user on the PG:



Share: \\192.168.215.2\Images (e.g. \\r123\Images)

User name: auduser (e.g. Domain\Username)

Password: *******

Figure 4-12 ServiceCenter: Add Network Drive

4. Define the computer name as an IP address; you cannot give it your own name.

   Share access to a directory or to the CD or DVD drive on the PG/PC must have been enabled.

4.4 Commissioning the replacement hard disk

Overview

The mechanical and electrical steps involved in replacing the PCU 50.3 hard disk are described in:

References: Operator Components and Networking Manual

Note

The replacement hard disk is delivered without the Windows operating system and without HMI system software.

The ServiceCenter incl. Symantec Ghost is installed at the factory on each PCU and on the replacement hard disk.

Creating a hard disk backup (disk image)

The entire contents of hard disks can be saved as a disk image file using the Symantec Ghost software. This "disk image" can be stored on various types of media for later restoration of the hard disk, e.g. on CD-ROM or a network drive.

Commissioning the replacement hard disk

After installing the replacement hard disk, the Ethernet interfaces of the PCU are preset in the following manner:

- Ethernet 1 (Company Network) as a standard DHCP client
- Ethernet 2 (System Network) as a SINUMERIK DHCP server with the fixed IP address 192.168.214.241 and subnet screen 255.255.255.0

The PCU must therefore be disconnected from the system network before the replacement hard disk is fitted.

To commission the hard disk, proceed as follows:

1. Connect a PG/PC as per the recommended configurations.
2. Start the ServiceCenter and select "Restore Disk Image".

Note

If the transfer is interrupted during the "Restore" process, no consistent system is available on the hard disk, i.e. the "Restore" process cannot be repeated because the PCU no longer boots up.

In this event, the "Emergency Boot System" is used.

See also

Requirements (Page 71)

How to backup and restore the hard disk (Page 67)

How to create a service system for PCU (Page 61)

Service and diagnostics 5

5.1 PCU Hardware Diagnostics

Purpose

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 50.3 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 errors without HMI Advanced

The PCU hardware monitor logs all hardware errors in the Windows event log so that the errors can be output even for a PCU without HMI software being installed.

The alarms are output in the log under "Control Panel" → "Administrative Tools" → "Event Viewer".

Note

The PLC interface of HMI Advanced is located in DB 10 of the PLC program and is supplied by the PCU hardware monitor in the event of a failure.

See also

Description of the alarms: Diagnostics Manual SINUMERIK

Operator Components and Networking Manual: Chapter PCU 50.3, spare parts

Commissioning the replacement hard disk (Page 80)

5.2 Evaluating 7-segment display

Purpose

The 7-segment display is intended for initial diagnosis of the PCU 50.3 during operations 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.

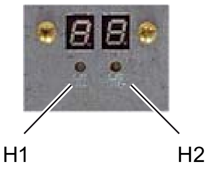


Figure 5-1 7-segment display with LEDs

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	
		0	1	Start Windows	Load device drivers needed for the Windows start
		0	2	PCU hardware service	The PCU hardware service has been started.
		0	5	Network	Wait for network interfaces to be ready.
		0	8	TCU support test step 1	Wait for FTP server to start
		0	9	TCU support test step 2	Wait for boot server to start for TCU network boot and for TCU hardware service to start

LED H1	LED H2	7-segment display		Status	
		0	A	TCU support test step 3	Wait for VNC server to start
		0	B	HMI manager	Wait for HMI manager to start

- During operation:

LED H1	LED H2	7-segment display		Status	
Flashing green	--	0	0	OK	
Flashing red	--			Error:	
		0	1	Temperature alarm	Housing or CPU temperature; temperature above limit value (SOM error)
		0	2	Fan alarm	Housing or CPU: Low fan speed or fan failure (SOM error)
		0	3	Hard disk alarm	S.M.A.R.T hard disk error
		0	6		Failure of VNC server or VNC server service stopped

- During shutdown

LED H1	LED H2	7-segment display		Status
Flashing red/green	--	0	5	Shutdown active

5.3 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)

5.4 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. NCUs or PCUs.

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.

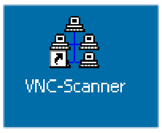
Furthermore, it enables you to call up a list of all the network devices, thereby creating an overview of the system network.

Note

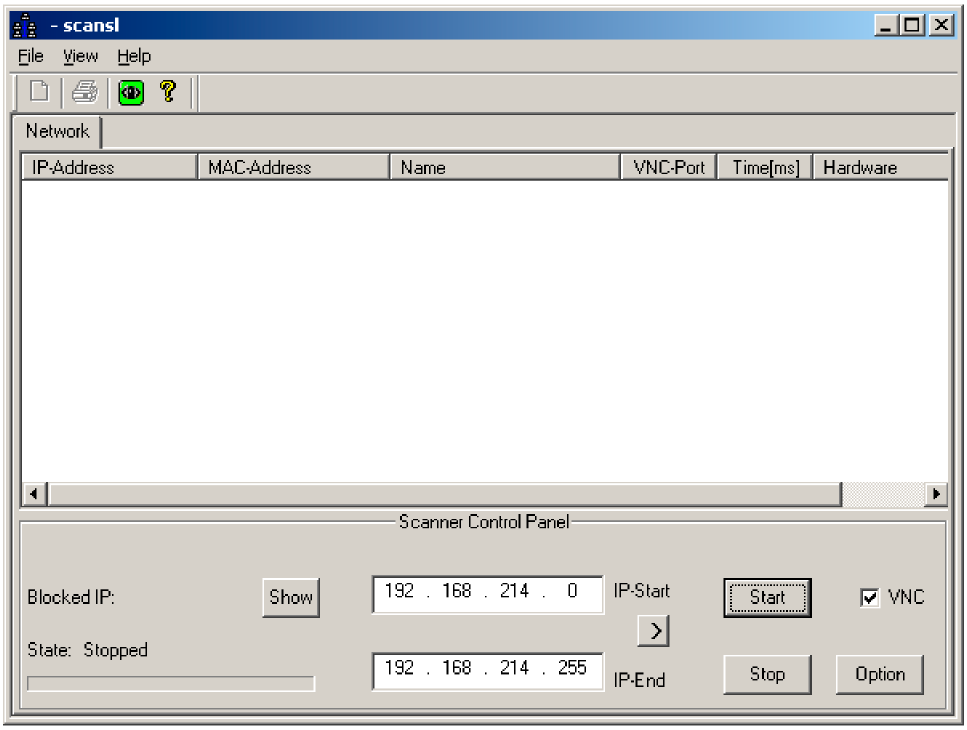
The VNC scanner is available on the Service Desktop of the PCU 50.3 and on the PCU-Basesoftware DVD for installation on a PG/PC.

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 50.3.



IP-Address	MAC-Address	Name	VNC-Port	Time[ms]	Hardware

Blocked IP:

State: Stopped

IP-Start: 192 . 168 . 214 . 0

IP-End: 192 . 168 . 214 . 255

☑ VNC

Figure 5-2 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.

3. "VNC" option:

   - ON: only search for VNC servers.
   - OFF: search all the stations in the network.

4. Start the search with the "Start" button.

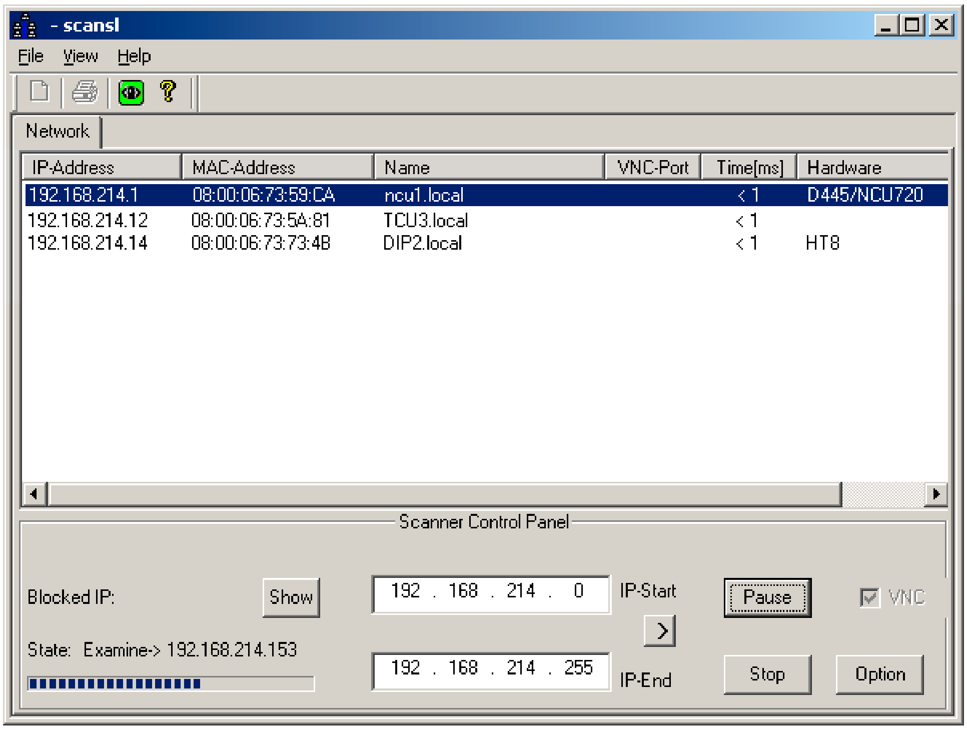
   The following station parameters will be listed:

   - 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.



IP-Address	MAC-Address	Name	VNC-Port	Time[ms]	Hardware
192.168.214.1	08:00:06:73:59:CA	ncu1.local		< 1	D445/NCU720
192.168.214.12	08:00:06:73:5A:81	TCU3.local		< 1	
192.168.214.14	08:00:06:73:73:4B	DIP2.local		< 1	HT8

Blocked IP:

State: Examine-> 192.168.214.153

IP-Start: 192 . 168 . 214 . 0

IP-End: 192 . 168 . 214 . 255

☑ VNC

Figure 5-3 VNC scanner: List of stations

5. 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).

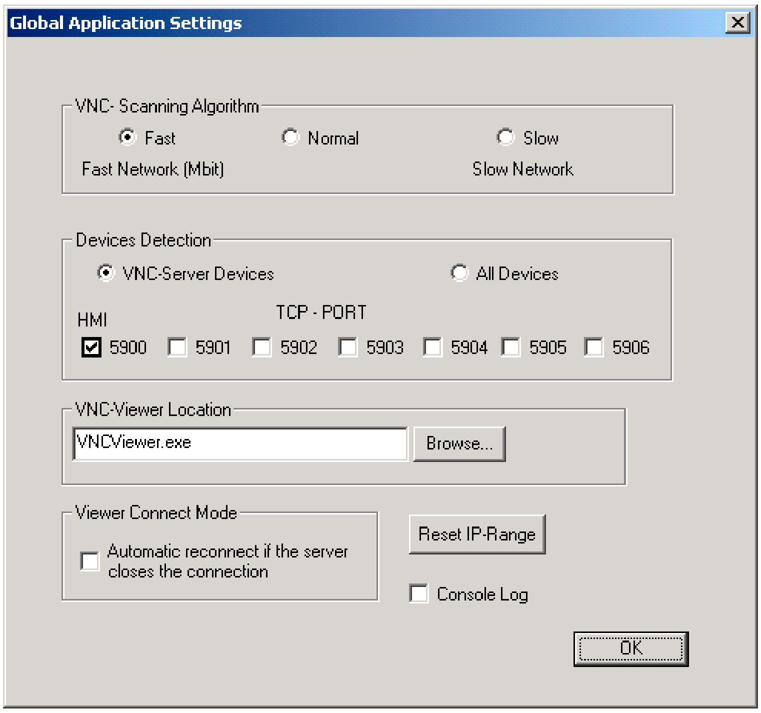
   To view the system from another station, this function must be enabled by the system. You can find the settings for this in the "System Network Center" or directly in tcu.ini in the [VNCViewer] section.

   Should you wish to access a PCU 50.3 via the company network using the VNC scanner or VNC viewer, port 5900 must be added on the PCU 50.3 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.



Global Application Settings

VNC- Scanning Algorithm
● Fast ○ Normal ○ Slow
Fast Network (Mbit) Slow Network

Devices Detection
● VNC-Server Devices ○ All Devices
HMI TCP - PORT
☑ 5900 □ 5901 □ 5902 □ 5903 □ 5904 □ 5905 □ 5906

VNC-Viewer Location
VNCViewer.exe [Browse...]

Viewer Connect Mode
□ Automatic reconnect if the server closes the connection

[Reset IP-Range]
□ Console Log
[OK]

Figure 5-4 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:

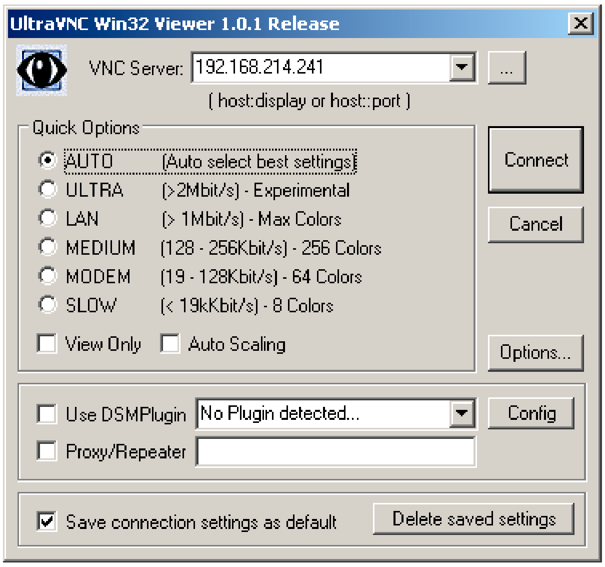
  Link to the VNC viewer (open source program: already pre-installed on the PCU; on a ServicePC this link must be specified).
- Viewer default connection mode:

  "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

CAUTION
Do not change defaults! To ensure that the VNC viewer works properly, the following options may not be changed.

After starting the VNC viewer, the following dialog opens:



UltraVNC Win32 Viewer 1.0.1 Release

VNC Server: 192.168.214.241
(host:display or host::port)

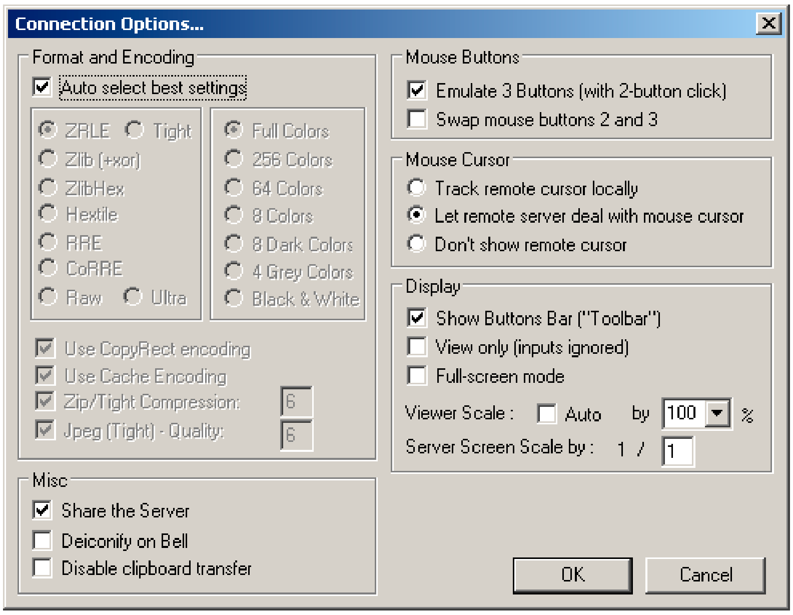
Quick Options
● AUTO (Auto select best settings)
○ ULTRA (>2Mbit/s) - Experimental
○ LAN (> 1Mbit/s) - Max Colors
○ MEDIUM (128 - 256Kbit/s) - 256 Colors
○ MODEM (19 - 128Kbit/s) - 64 Colors
○ SLOW (< 19kKbit/s) - 8 Colors
□ View Only □ Auto Scaling

□ Use DSMPlugin No Plugin detected...
□ Proxy/Repeater

☑ Save connection settings as default

Figure 5-5 Default: UltraVNC

After clicking the "Options ..." button, the following dialog opens:



Connection Options...

Format and Encoding

☑ Auto select best settings

● ZRLE ○ Tight
○ Zlib (+xor)
○ ZlibHex
○ Hextile
○ RRE
○ CoRRE
○ Raw ○ Ultra

● Full Colors
○ 256 Colors
○ 64 Colors
○ 8 Colors
○ 8 Dark Colors
○ 4 Grey Colors
○ Black & White

☑ Use CopyRect encoding
☑ Use Cache Encoding
☑ Zip/Tight Compression: 6
☑ Jpeg (Tight) - Quality: 6

Misc

☑ Share the Server
□ Deiconify on Bell
□ Disable clipboard transfer

Mouse Buttons

☑ Emulate 3 Buttons (with 2-button click)
□ Swap mouse buttons 2 and 3

Mouse Cursor

○ Track remote cursor locally
● Let remote server deal with mouse cursor
○ Don't show remote cursor

Display

☑ Show Buttons Bar ("Toolbar")
□ View only (inputs ignored)
□ Full-screen mode

Viewer Scale : □ Auto by 100 %

Server Screen Scale by : 1 / 1

Figure 5-6 Default: Connection Options ...

5.5 OpenSSH for WinSCP and PuTTY

Intended use

The service tools WinSCP and PuTTY are already used to access NCU 7x0. They can now similarly be used to access the PCU50.

Default

The user "auduser" is enabled for SSH access via its login name + password. Preferred SSH clients are PuTTY or WinSCP.

Function

The COPSSH tools "activate user" and "deactivate user", including documentation, are accessible from the service desktop via "Start" → "Programs" → "Service Tools" → "COPSSH".

When the user logs in via SSH, his home directory, under which access takes place, is set as the current directory. By changing to the ../../cygdrive directory, the user can branch to the c: d: e: and f: drives.

See also:

"NCU Operating System" Commissioning Manual (IM7):
Description and Licenses for WinSCP and PuTTY.

List of Abbreviations A

A.1 Abbreviations

CF	CompactFlash card: Memory card
DCK	Direct Control Keys Direct control key
DCP	Device Control 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: 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: PLC
RAM	Random Access Memory: Program memory which can be read and written into
RDY	Ready Ready
SNMP	Simple Network Management Protocol (network protocol for monitoring and controlling network elements such as routers, servers, switches, and printers from a central station)
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

Appendix A

A.1 Feedback on the documentation

This document will be continuously improved with regard to its quality and ease of use. Please help us with this task by sending your comments and suggestions for improvement via e-mail or fax to:

E-mail:	mailto:docu.motioncontrol@siemens.com
Fax:	+49 9131 - 98 2176 Please use the fax form on the back of this page.

To SIEMENS AG I DT MC MS1 P.O. Box 3180 D-91050 Erlangen / Germany	From Name:
	Address of your company/department
	Street:
	Zip code: City:
	Phone: /
Fax: +49 9131 - 98 2176 (Documentation)	Fax: /

Suggestions and/or corrections