SIEMENS

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 7.5

01/2008
6FC5397-0DP10-2BA0

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
90327 NÜRNBERG
GERMANY

Ordernumber: 6FC5397-0DP10-2BA0
℗ 12/2007

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

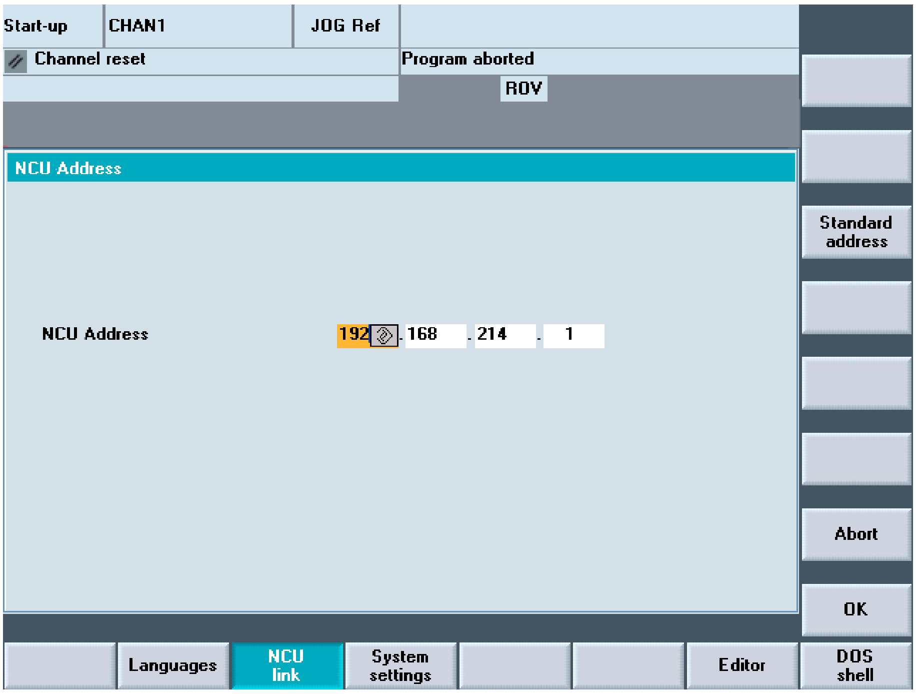
Languages

Using this function, you select the first and second language; you can toggle using the "Change Language" softkey between the following standard languages:

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

NCU link

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



NCU Address 192 . 168 . 214 . 1

Figure 1-1 IP address of the NCU

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

Creating user-specific alarm texts (Page 81)

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 21)

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

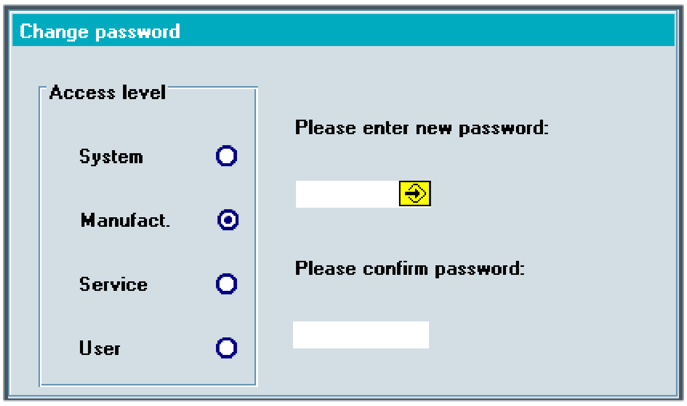
The operator only has access to information protected by this particular access level and the levels below it. The machine data is assigned different access levels by default.

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

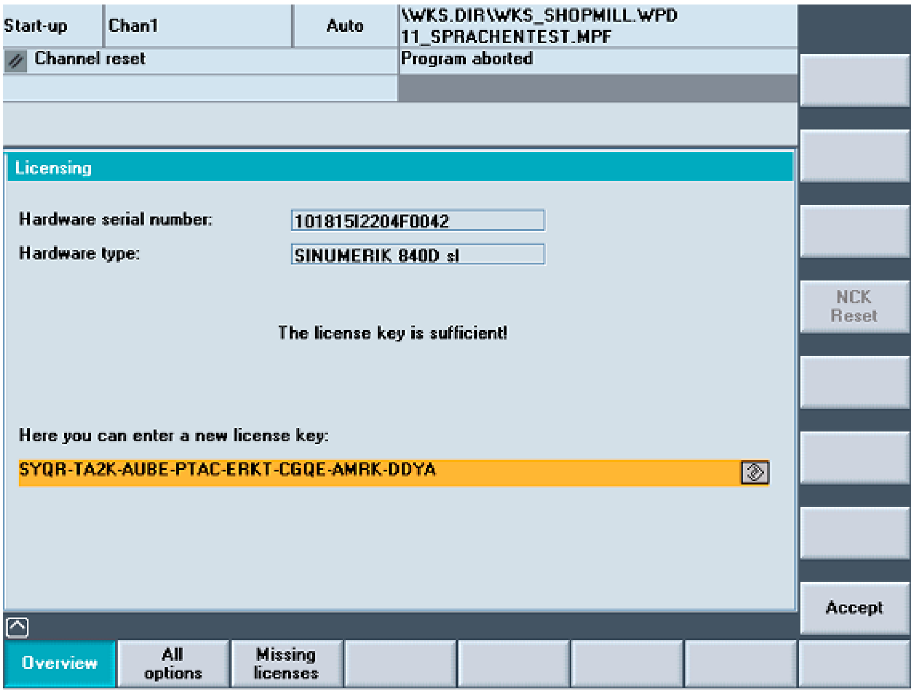
8. The modified password must be entered in both input fields then confirmed by pressing the "OK" softkey. Both entered passwords must match for the modified password to become valid.

1.4 Licensing

Licensing an option

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

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 Summary, licenses

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.

Configuring the HMI system 2

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.

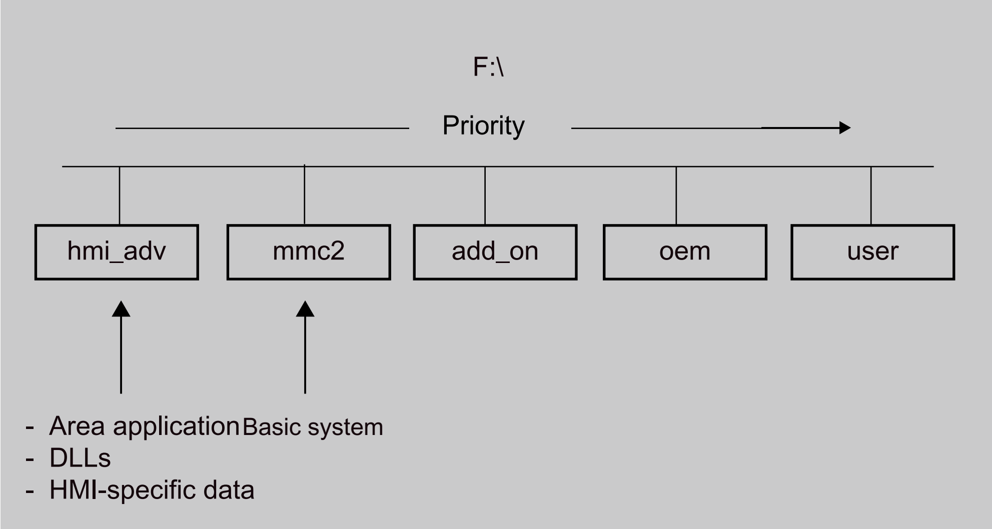


Figure 2-1 Priority

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)

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

Maximum value: 32000

With default value of 150, 75 alarms can be displayed. 2 records are needed for each alarm.

Selecting alarms by feature

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

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

Expression expresses selection features and is structured as follows:

Syntax

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

- IDENTIFIER:

No.	Alarm number
Prio	Priority
Mode	Message line/alarm line or dialog box
Type	Alarm type (PowerOn, Cancel, ...)
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 POWERON and RESET alarms are logged.

Filter=From:NCU_1

Only alarms from NCU_1 are logged

Filter=From:NCU_1 Type:1,3

Only POWERON and CANCEL alarms of the NCU_1 are logged

See also

MBDDE.INI (Page 281)

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 authorization levels for programs

Function

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

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

Restrictions

When the HMI database is installed, data are always created with the access authorizations defined in the standard data scheme. Access authorizations 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 267)

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.

x	y	Explanation
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*.
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 106)

TASKCONF.INI (Page 306)

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 266)

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.

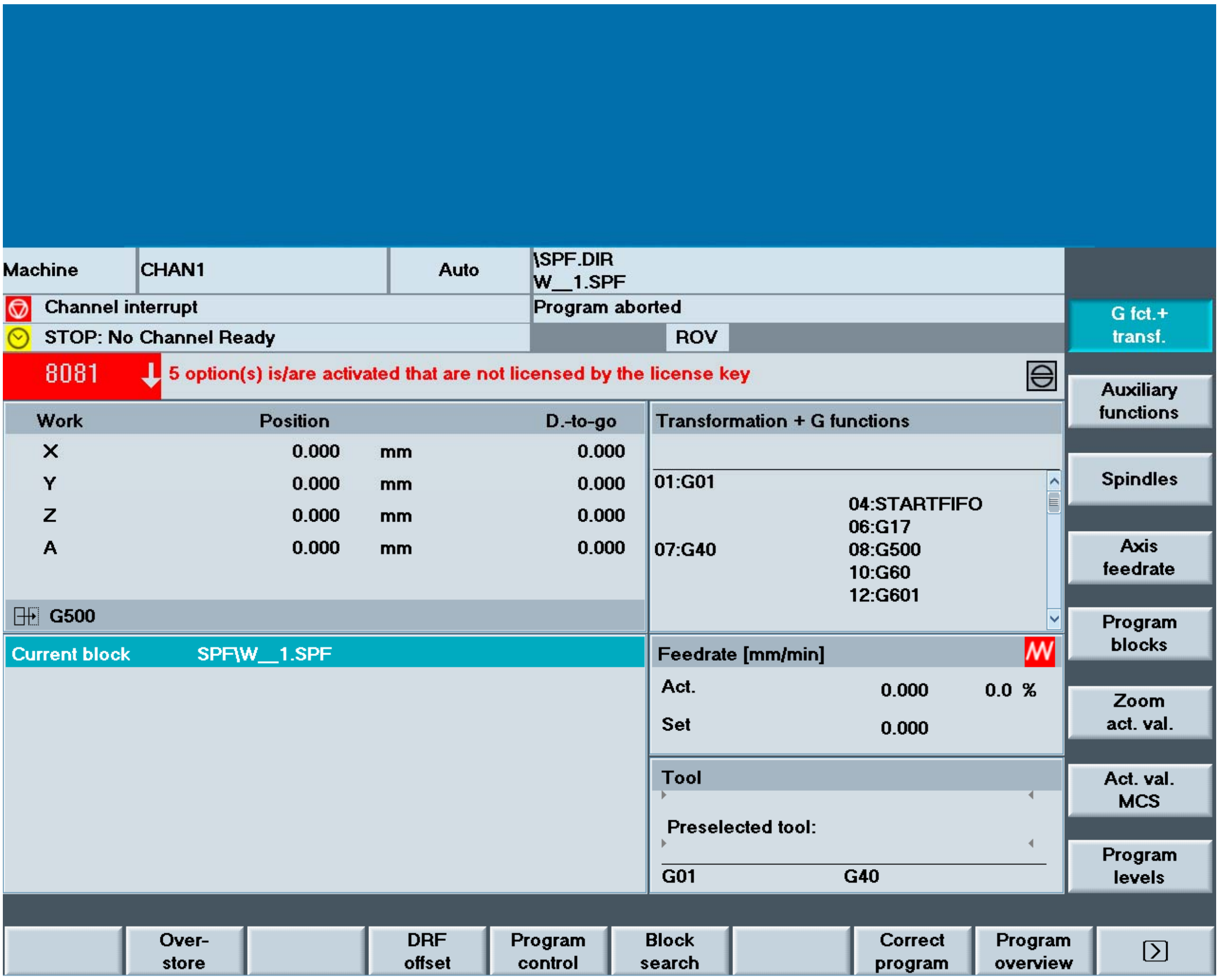
  Procedural control stores the start position of the HMI program in the HMI_Start entry in section [CONTROL] in the MMC.INI file.
- OEM applications that wish to use Wide Display must be regenerated with the current OEM package and the expansion of procedural control.

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

Transformation + G functions

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

Current block SPF\W__1.SPF

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\lfm_.com

First of all, the non-language-specific text file "lfm.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. "lfm_gr.com") is opened.

Extract from Short Help txt file "lfm_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 274)

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!
```

See also

Additional configuring possibilities for multi-channel step sequence programming:

MMC.INI (Page 285)

SEDITOR.INI (Page 297)

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".

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

See also

NETNAMES.INI (Page 290)

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 264)

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

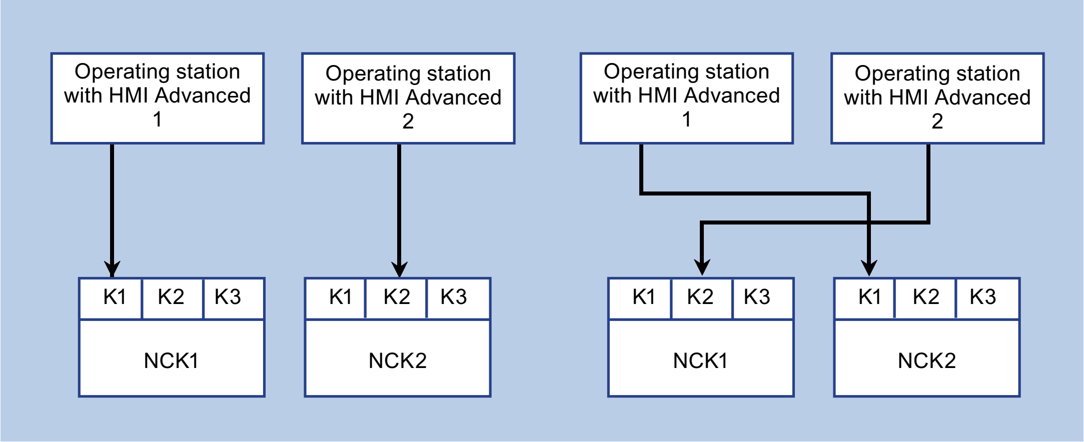

Operating station with HMI Advanced 1
Operating station with HMI Advanced 2
Operating station with HMI Advanced 1
Operating station with HMI Advanced 2
K1
K2
K3
NCK1
K1
K2
K3
NCK2
K1
K2
K3
NCK1
K1
K2
K3
NCK2

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

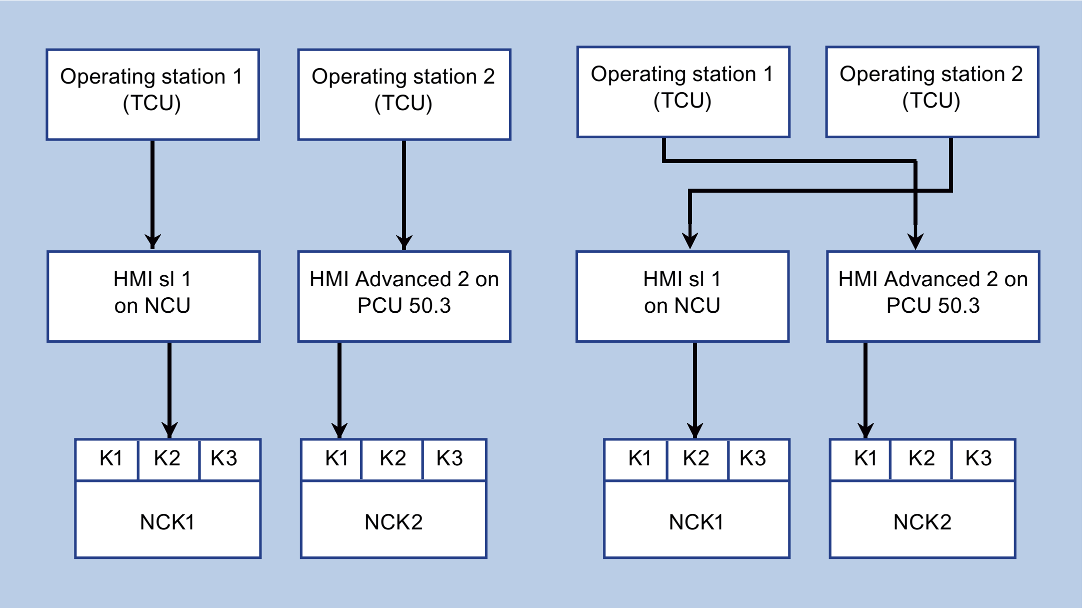

Operating station 1 (TCU)
Operating station 2 (TCU)
Operating station 1 (TCU)
Operating station 2 (TCU)
HMI sl 1 on NCU
HMI Advanced 2 on PCU 50.3
HMI sl 1 on NCU
HMI Advanced 2 on PCU 50.3
K1 K2 K3
NCK1
K1 K2 K3
NCK2
K1 K2 K3
NCK1
K1 K2 K3
NCK2

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.

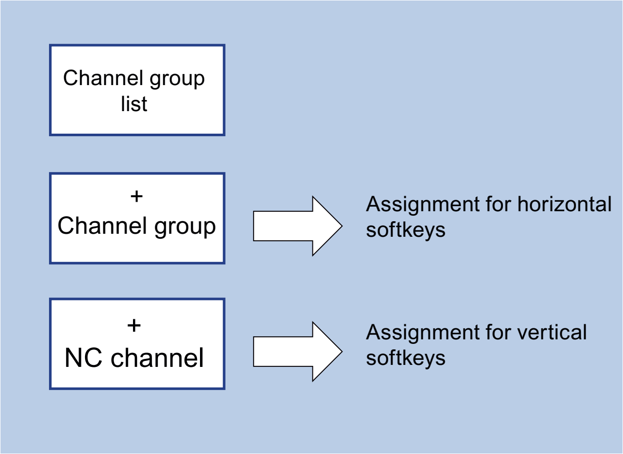

Channel group list
+
Channel group
Assignment for horizontal softkeys
+
NC channel
Assignment for vertical softkeys

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
```

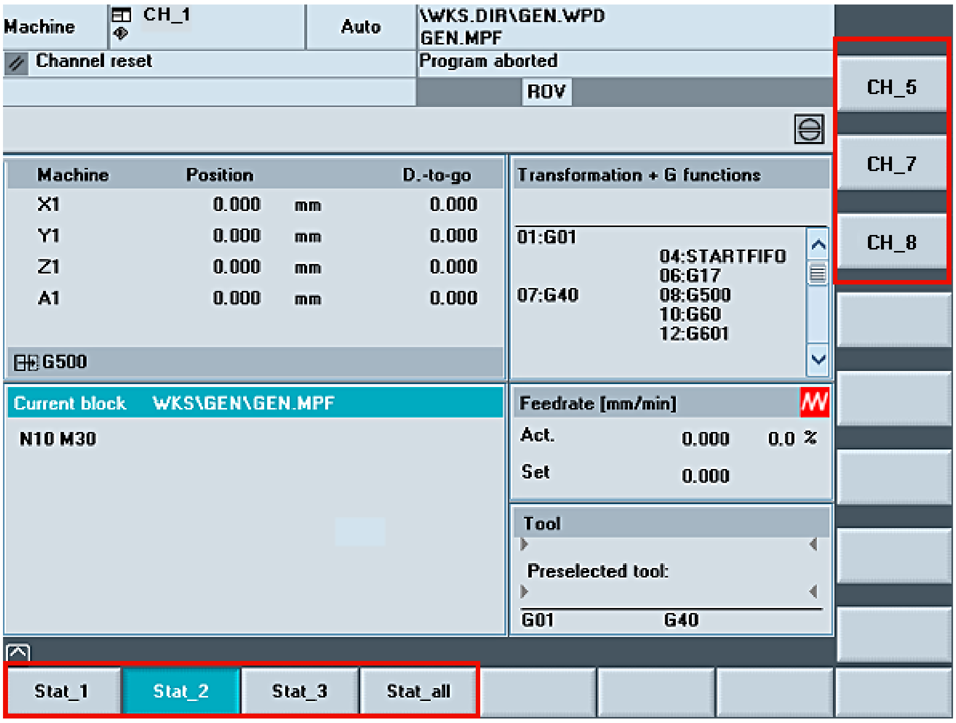
```
[Stat_2]
logChanList = CH_5, CH_7, CH_8

[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:



Machine	Position		D.-to-go
X1	0.000	mm	0.000
Y1	0.000	mm	0.000
Z1	0.000	mm	0.000
A1	0.000	mm	0.000

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 290)

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:

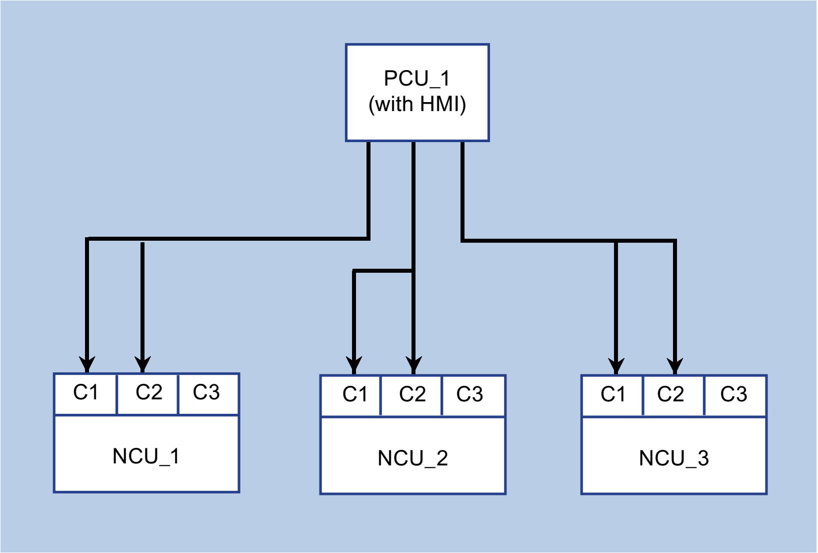


Figure 2-7 Switchover options

See also:

- On configuring the channel menu for the application case: NETNAMES.INI (Page 290)
- On displaying the channel menu in HMI Advanced: Configuration of a channel menu for direct channel selection (1:1 configuration) (Page 50)

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 50)

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:

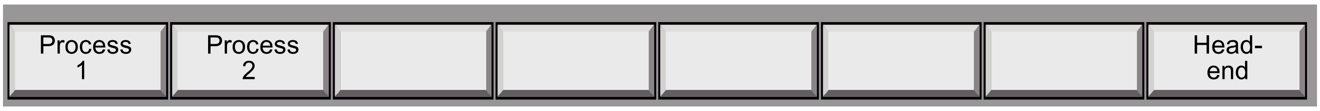


Figure 2-8 Horizontal softkey menu

2.3 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 275)

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 281)

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).

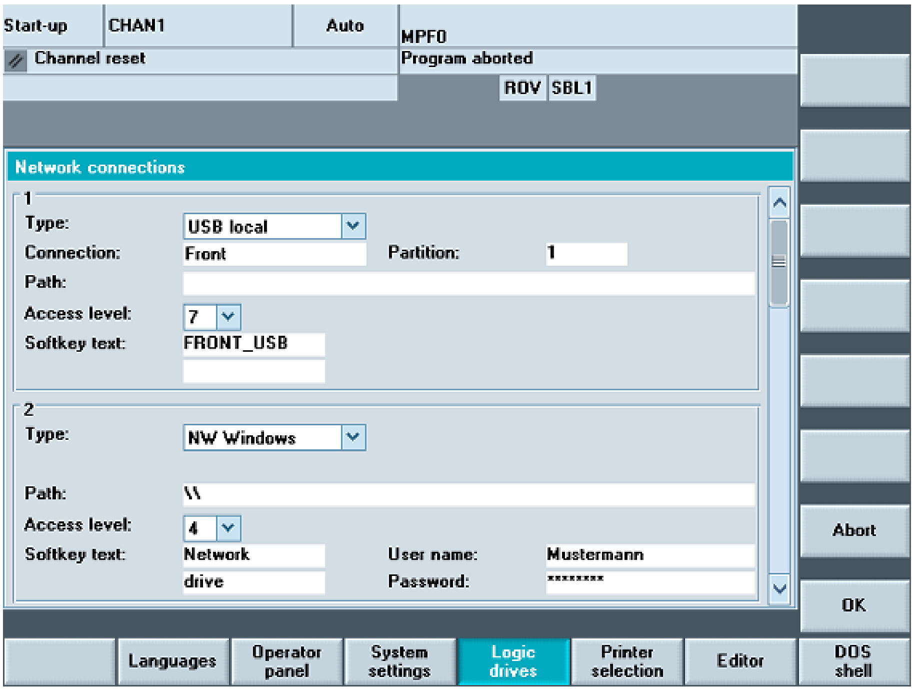
Softkey "USB local"

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. 2
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. tcu
NW Windows:	Designation of the network drive (under Windows). The path must begin with // or \\.
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:

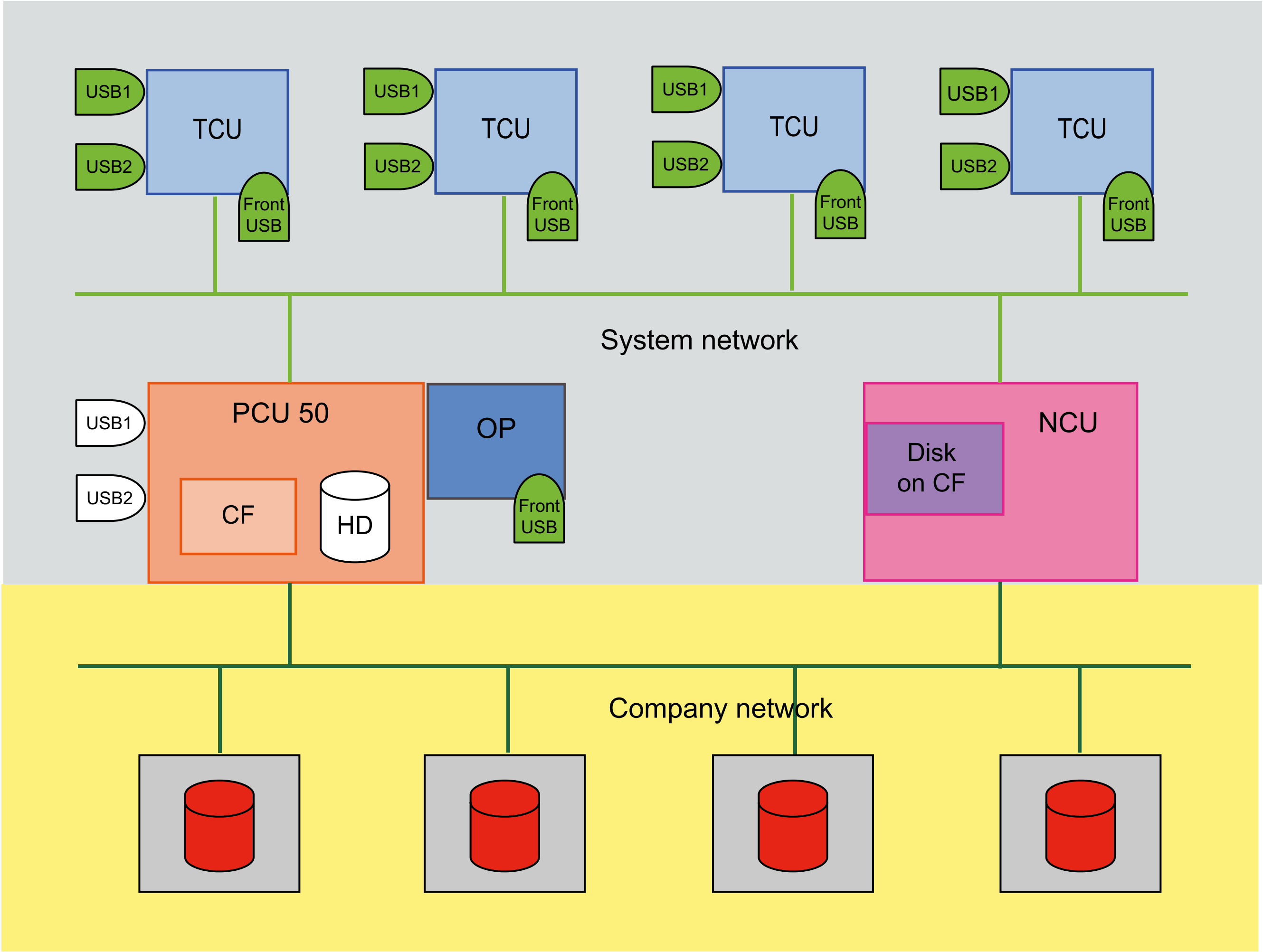


Figure 2-10 Network drives option

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

2.4 Optimizing 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

User programs

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 Creating user alarms

2.5.1 Structure of user-specific alarms

Classification of alarms

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		

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 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.

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

Supported languages

Table 2-2 Languages and ANSI tables / code pages used

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

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)

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.

Language

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.

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 26)

2.5.2 Creating user-specific alarm texts

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, "X" 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.

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 281)

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.

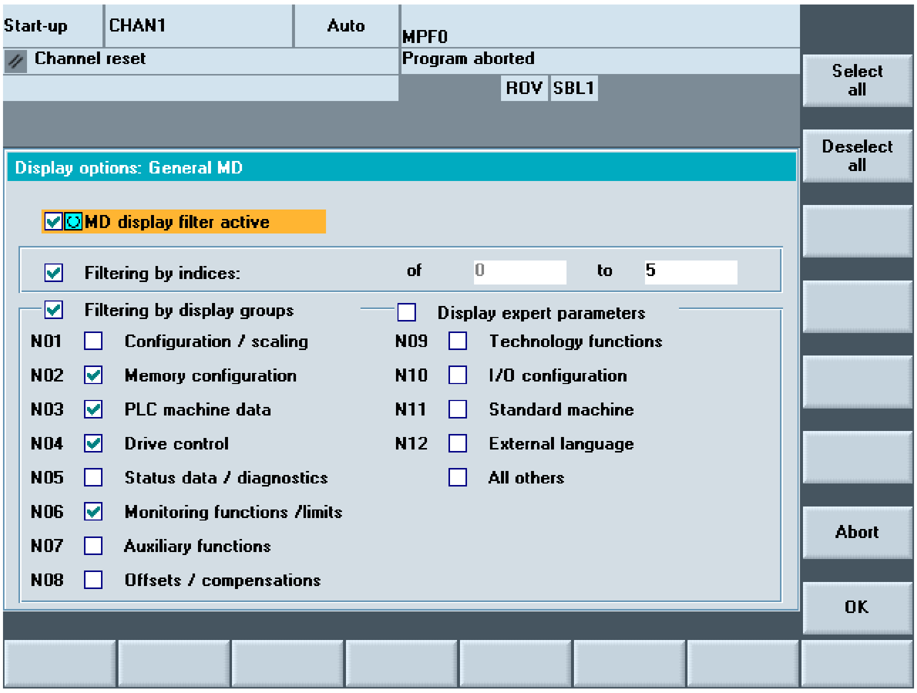
For this purpose, all machine data are assigned to certain display groups, e.g. configuration data. The following applies:

- 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:

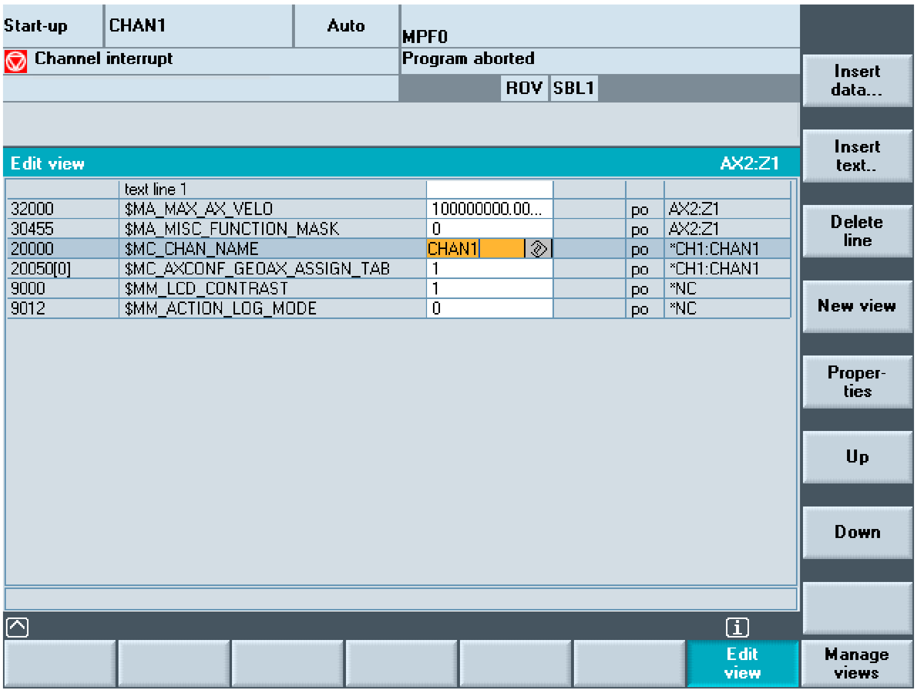
- General machine data
- Channelspecific machine data
- Axis-specific machine data
- General setting data
- Channel-specific setting data
- Axis-specific setting data
- General display machine data
- Channel-specific display machine data
- Machine data for SIMODRIVE drives (powerline)
- Machine data 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:



Channel interrupt

Program aborted

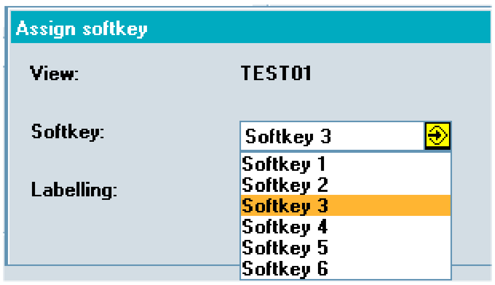
Edit view AX2:Z1

	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 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.

3.2.2 Machine data for setting actual value, scratching, PRESET

Function

The behavior of the **Scratching** and **Set actual value, PRESET** functions is influenced by a number of machine data. The selected settings also affect operation. (Display of softkeys and values, saving of input values).

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
	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: The old PRESET in the machine operating area (default)

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.

Reference: Operating Manual HMI-Advanced

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.

3.2.3 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)

3.2.4 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'
```

3.2.5 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)

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.

3.2.6 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".

3.2.7 Change access authorization for softkey "Basis WO"

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.

3.2.8 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

20098	DISPLAY_AXIS		
	Bit 3	= 1 = 0	Display geometry axis in handwheel selection window Hide geometry axis in handwheel selection window
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

3.2.9 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.

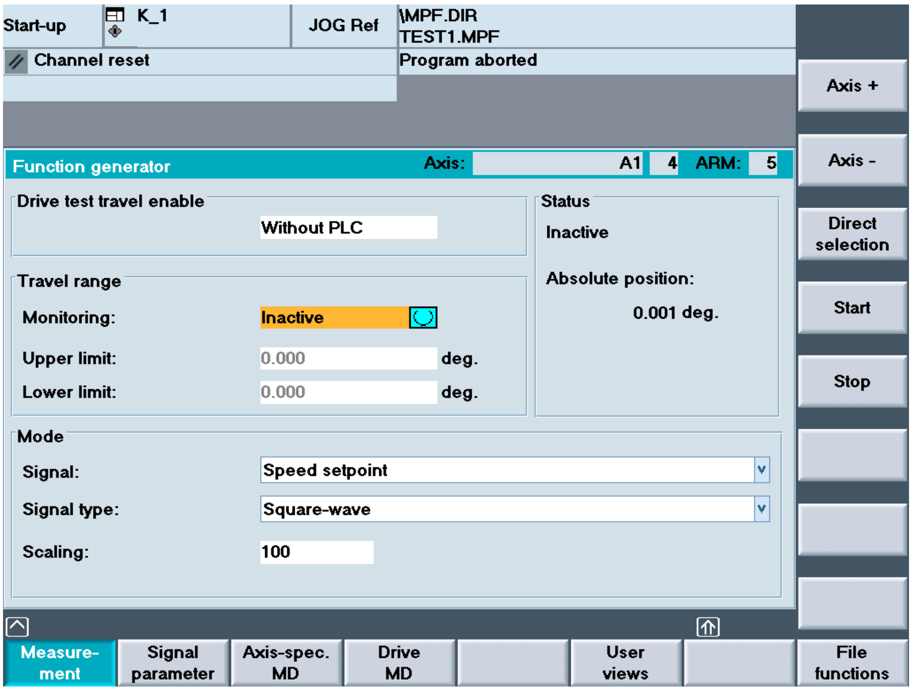
9428	$MM_MA_SPIND_MAX_POWER		
MD number	Maximum value of the spindle power display		
Default setting: 100		Min. input limit: 100	Max. input value: ****
Changes effective after: POWER ON		Protection level: 3/4	Units: %
Data type: WORD			Applies from SW version: 6.4
Significance	In this MD, enter the factor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

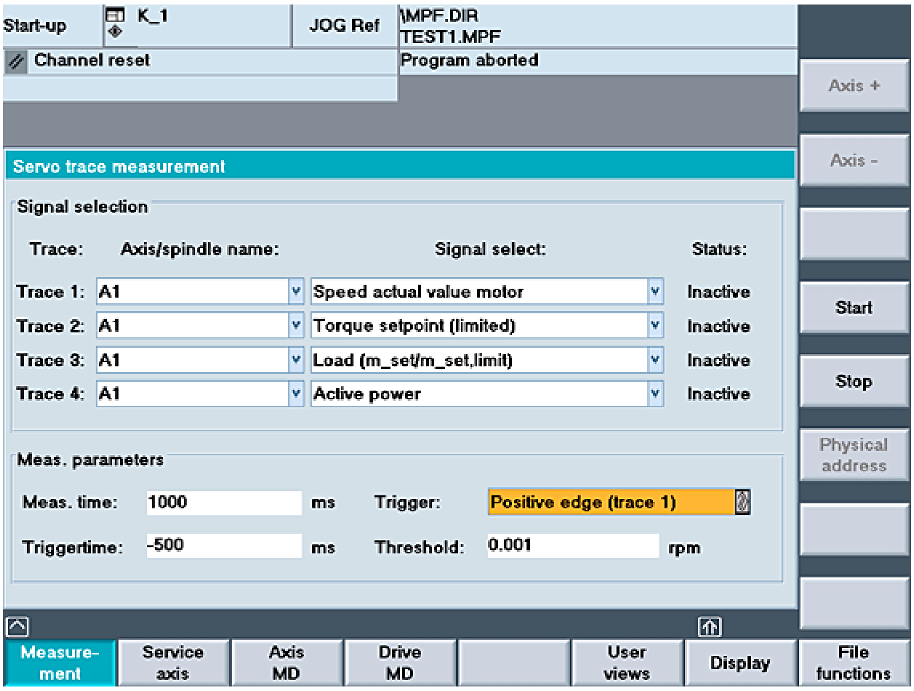
Mode
Signal: Speed setpoint
Signal type: Square-wave
Scaling: 100

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



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.

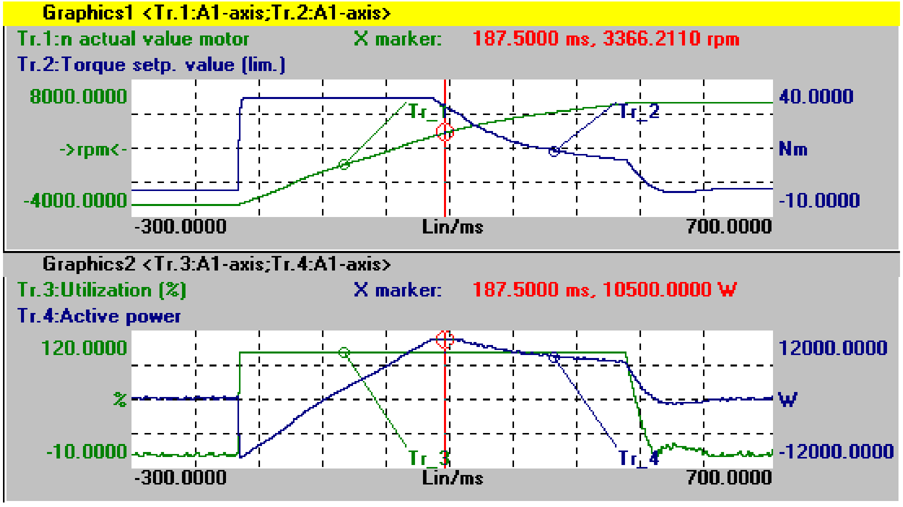


Figure 3-6 Servo trace: Evaluation

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.

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.	

3.3 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 operating messages 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

Structure of user-specific alarms (Page 78)

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

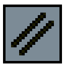
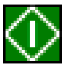
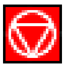
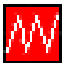
The status of the following components can be output with icons in the status line (in the header):

- 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.

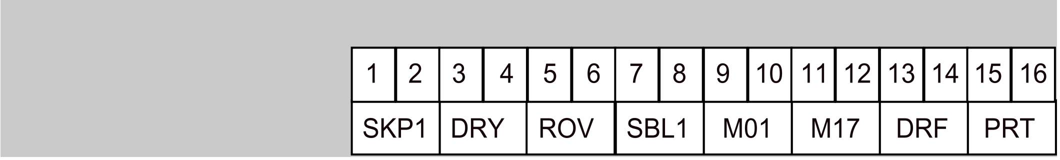
Note

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:

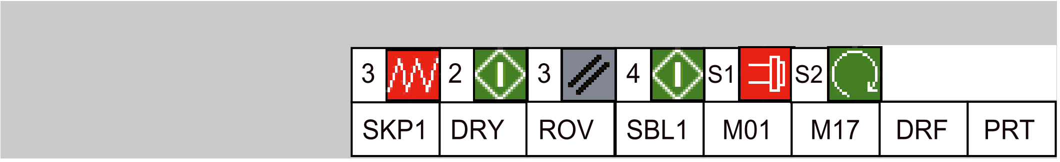


Figure 4-2 Program status line with icons (1)

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:

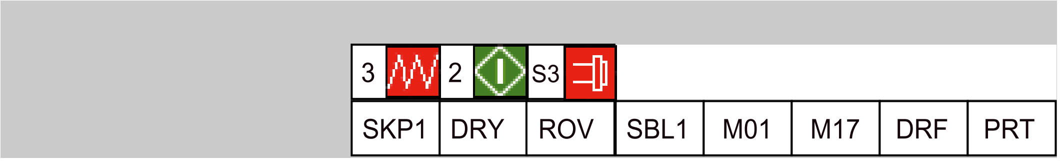


Figure 4-3 Program status line with icons (2)

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 268)

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.

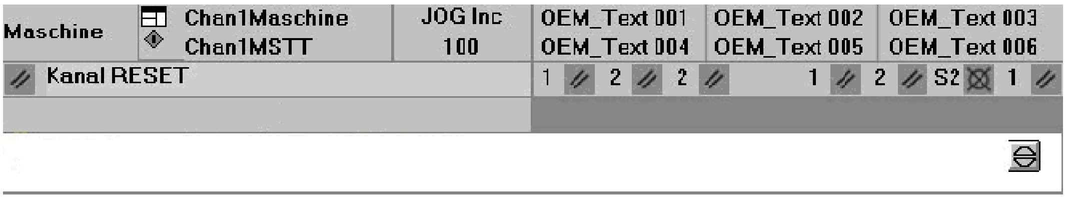


Figure 4-4 Example for OEM text

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>
```

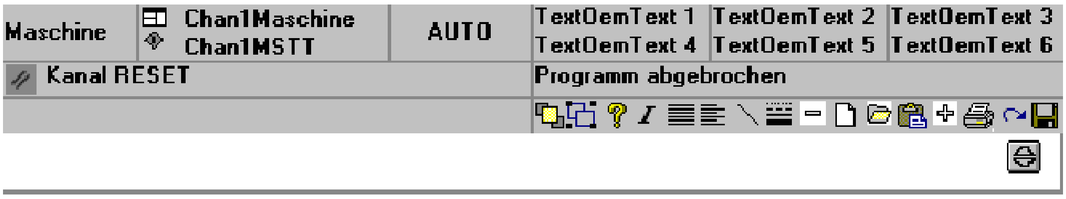
Here, <name> is the file name of the bit map and <Position> is the output position (from 1 to 16) in the display line. Multiple bit maps can be output at the same position. If more than one bit map is active simultaneously at the same position, the bit map with the highest screen number is displayed.

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 268)

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.

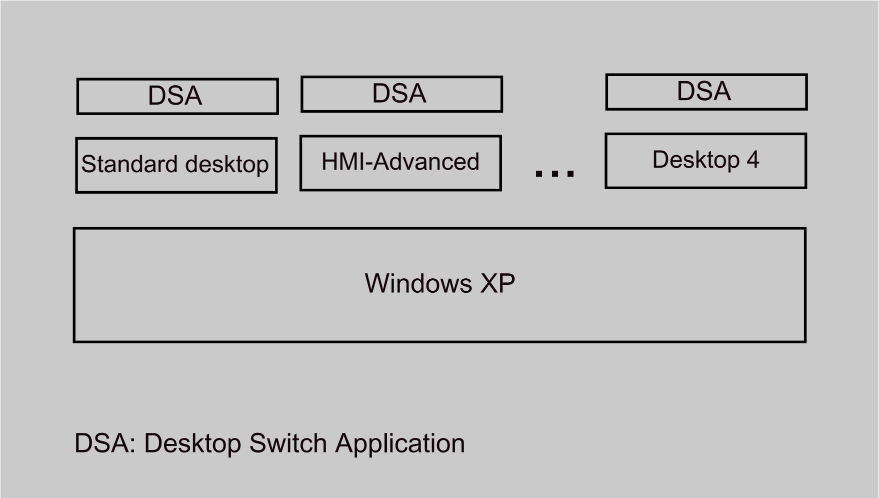


Figure 5-1 Separate desktops for HMI-Advanced and other applications

---

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:

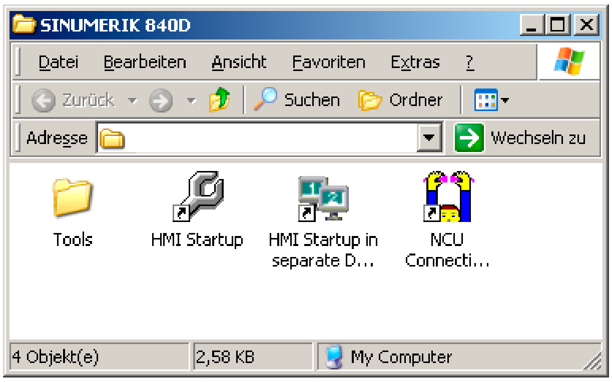
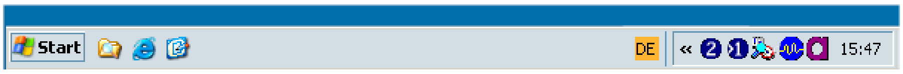


Figure 5-2 Linking to the desktop switching application

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

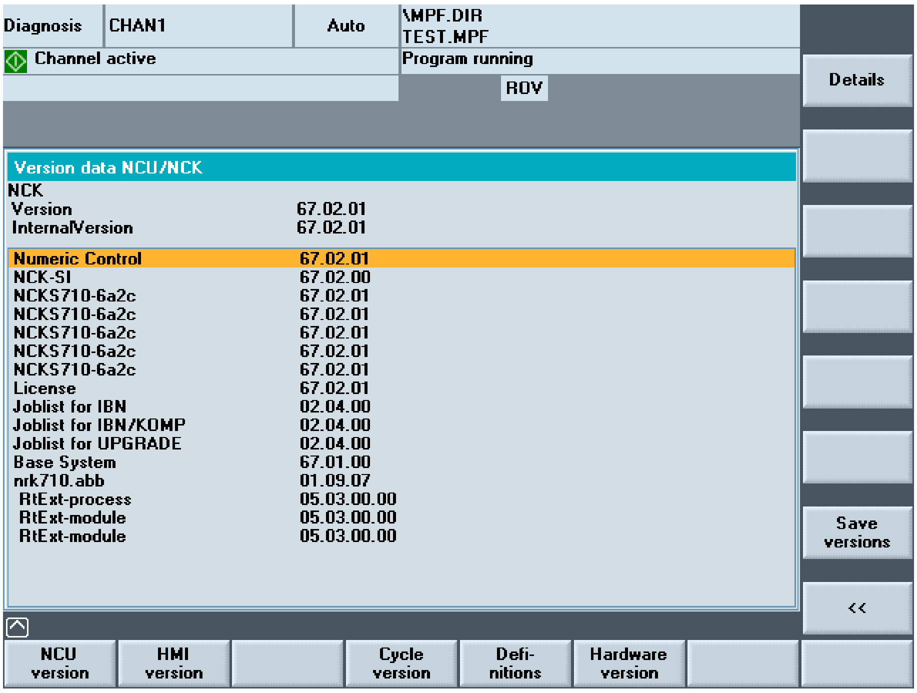
Overview

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.

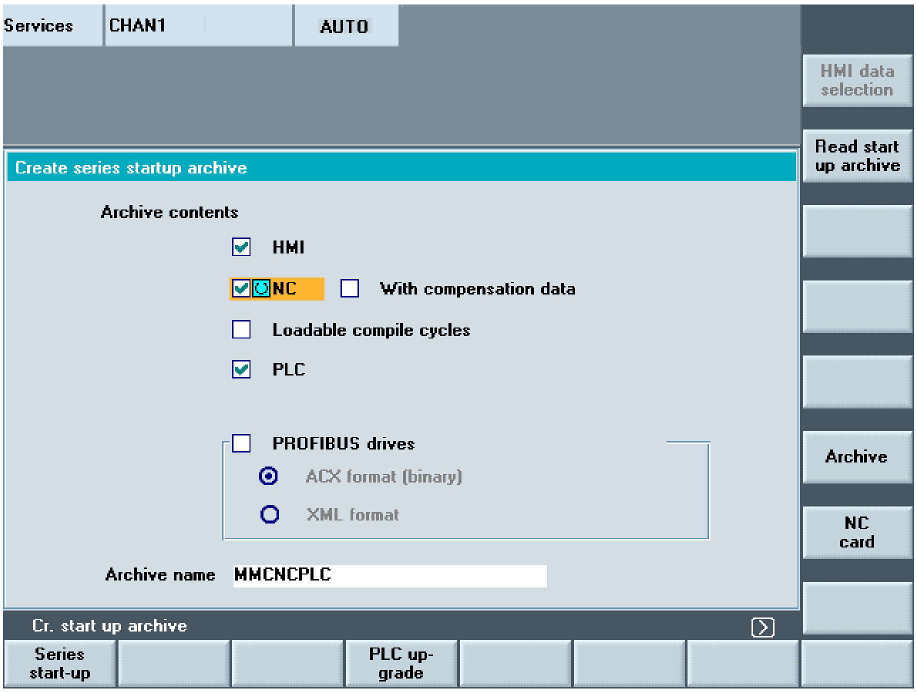
The drive data cannot be changed and can be optionally saved in the acx format as binary data or in the xml format.

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:

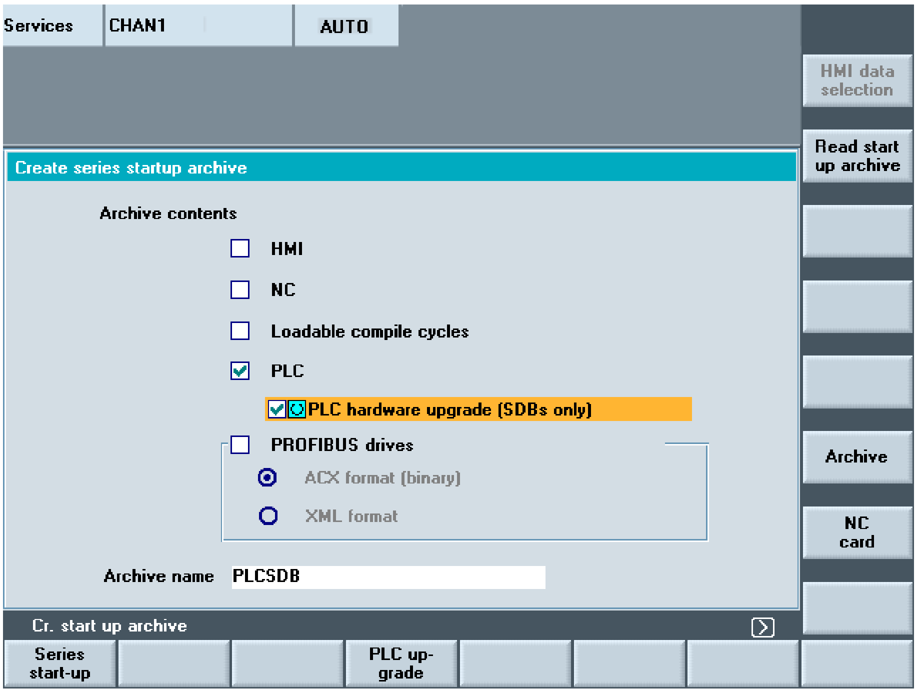
- Upgrading the PLC

  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

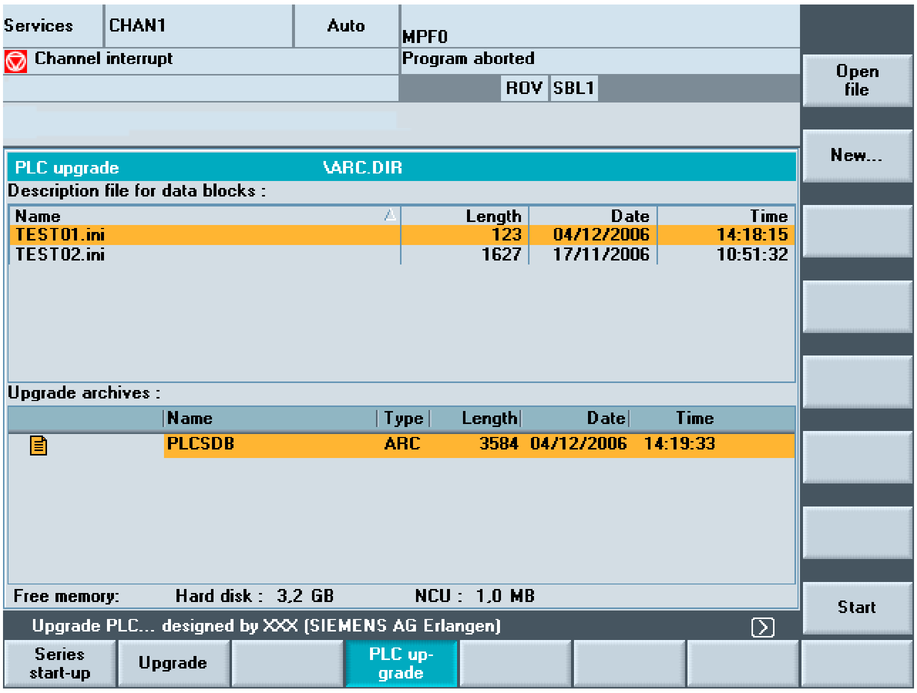
Procedure

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

ROV SBL1

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.

Communications error log

Errors that occur during communication between the HMI and NCK/PLC are registered in a communication error log. This protocol is displayed when the "Comm protocol" softkey is pressed.

The error log file is principally used by the control manufacturer (Siemens) to diagnose communication errors.

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.

5.5.4 Communication error log

Function

Errors that occur during communication between the HMI and NCK/PLC are registered in a communication error log. You can display the errors by pressing this softkey.

The error log file is principally used by the control manufacturer (Siemens) to diagnose communication errors.

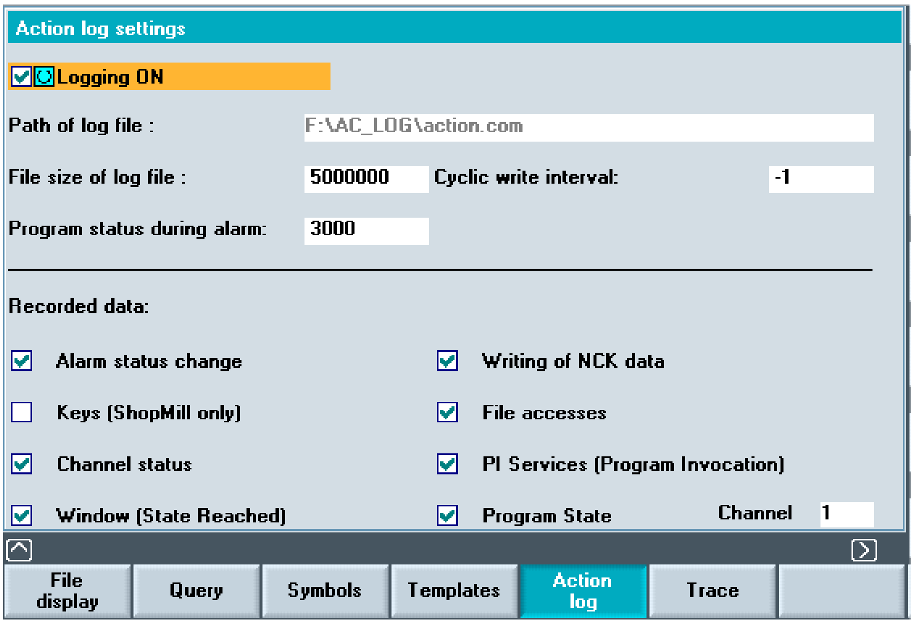
5.6 Action log

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.
- Program state / channel: **(from NCK Version 7.4)** 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 262)

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 (same as, e.g. key lock, screensaver control, etc.). 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.

Select e.g. "Services" → "Data Out" → "Disk", in order to read-out the log file.

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]

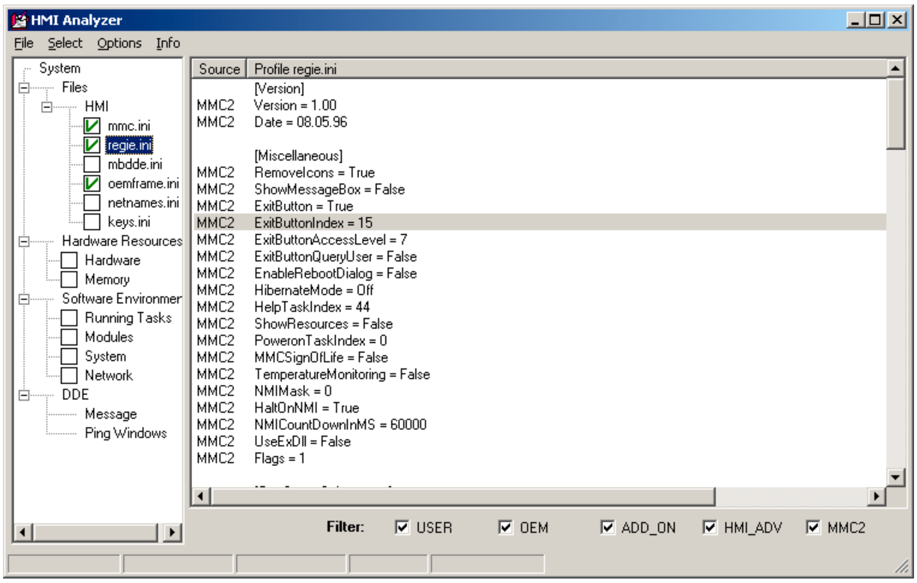
Additional INI files that are not subject to the INI concept of HMI-Advanced, i.e. INI files from add-on applications, can be entered here. The complete path is needed for the entries. Only the file of this path is analyzed.

5.7.2 Using the HMI Analyzer

Overview

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 5-8 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.

For 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)

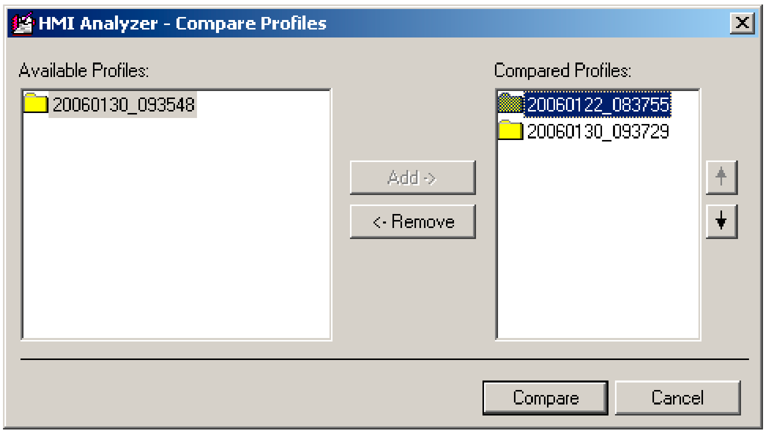
Individual profiles can be deleted from the hard disk via the menu item "File" → "Delete Profile".

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.



Available Profiles: 20060130_093548

Compared Profiles: 20060122_083755, 20060130_093729

Figure 5-9 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:

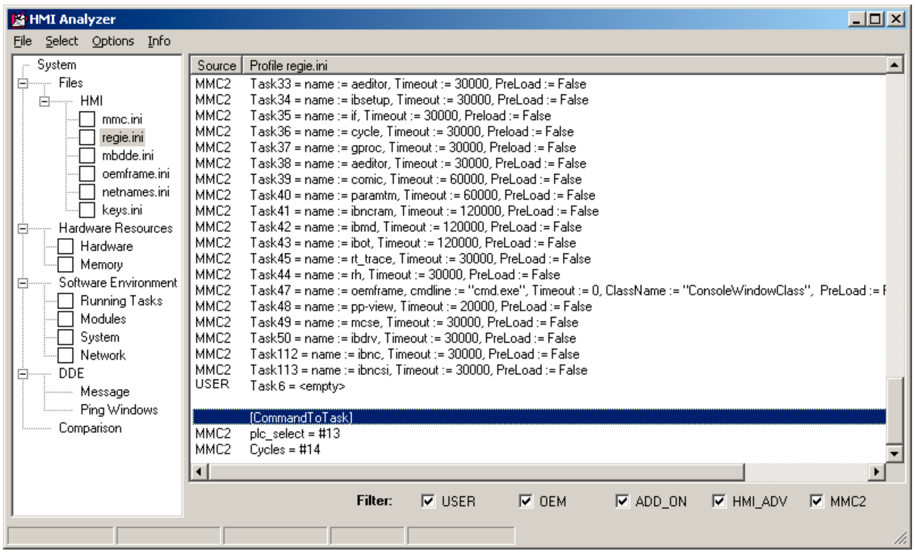
Due to the hierarchical execution of the HMI directory, the entry made in the Regie.ini file was overwritten by an entry in another regie.ini file with a higher priority and is therefore not in effect.

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 5-10 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:

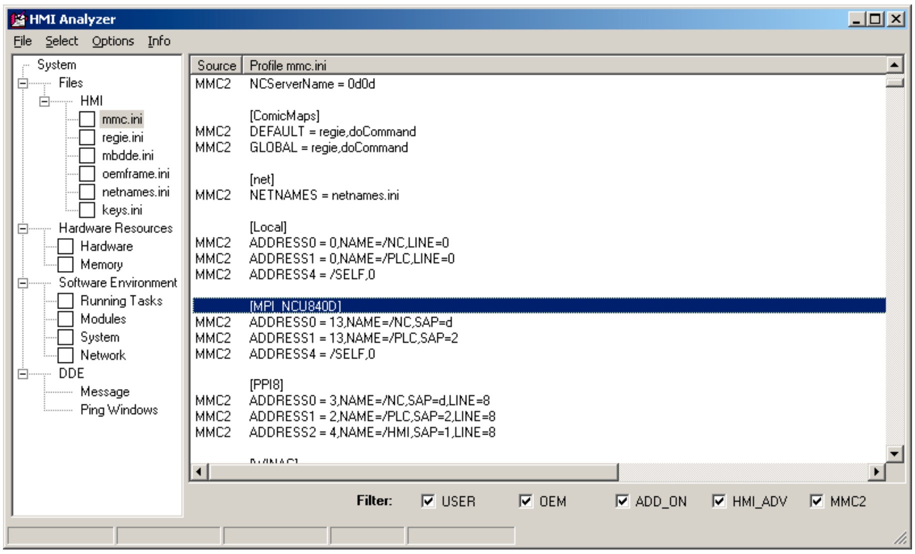
Due to the hierarchical execution of the HMI directory, the entry made in the mmc.ini file was overwritten by an entry with a higher priority and is therefore not in effect.

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 5-11 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.

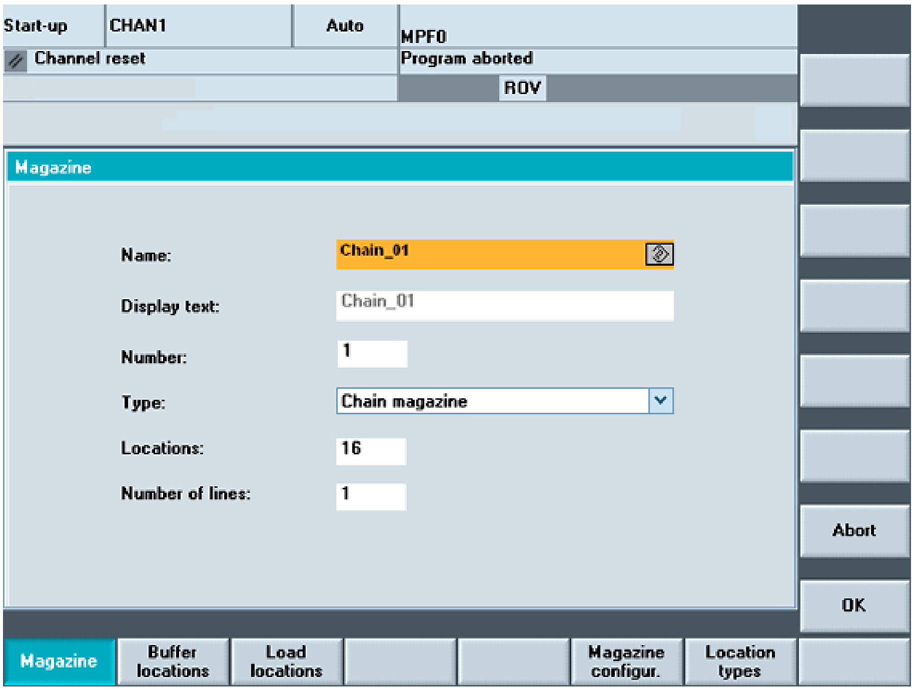
Tool management 6

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.



Start-up | CHAN1 | Auto | MPF0

Channel reset

Program aborted

ROV

Magazine

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

Magazines	
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.

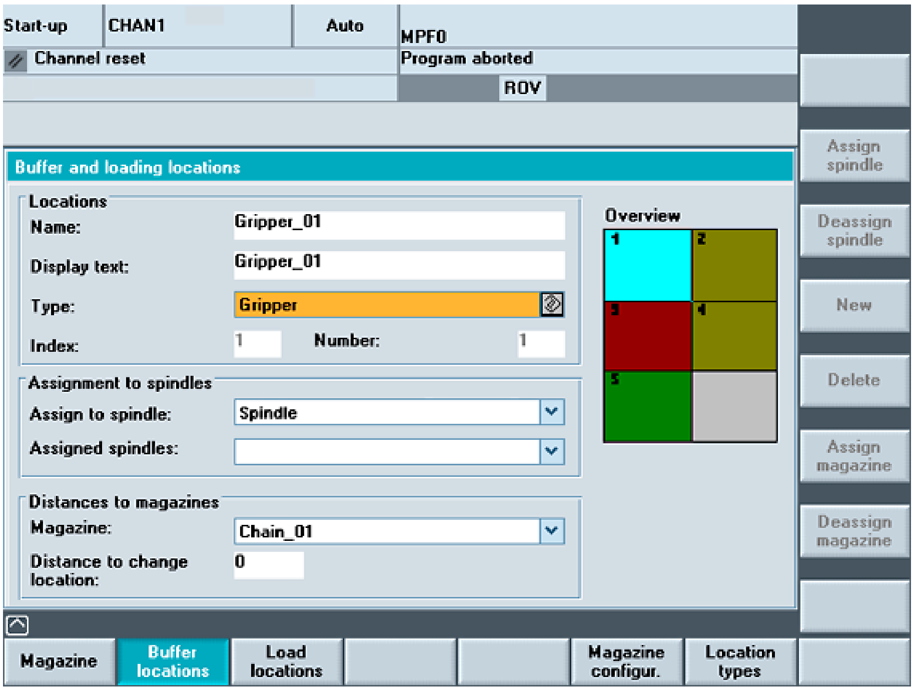
> Note
>
> The next time the PLC is booted, the message "Deleting DBxx in PLC..." may appear. In this case, the specified DB is deleted via STEP 7. As an alternative, a suitable PLC archive can also be loaded.

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.

Locations	
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:

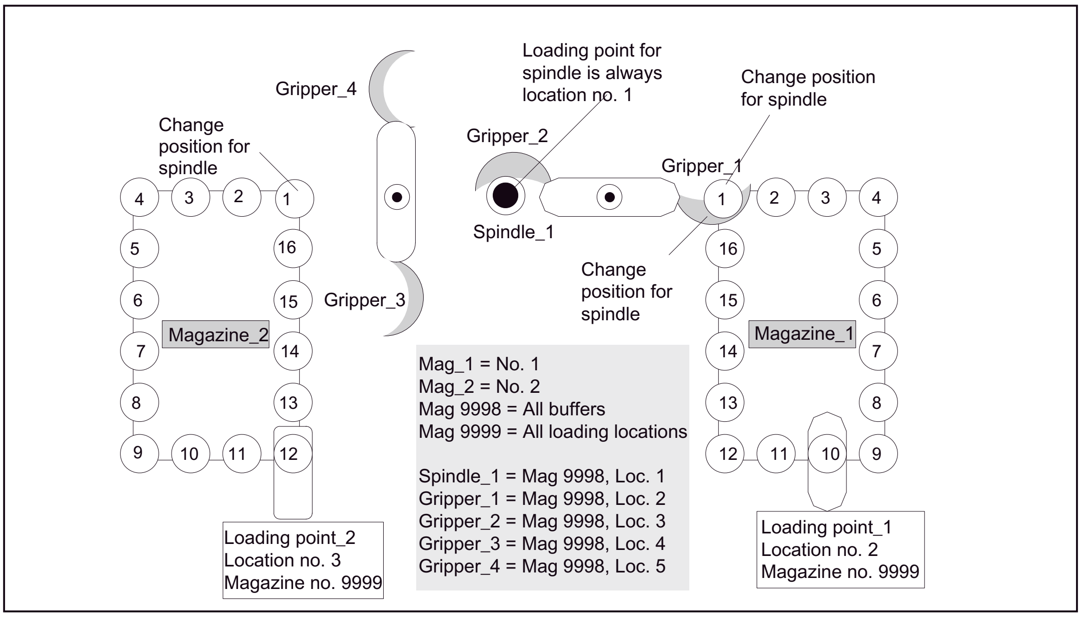
1. Press the "New" softkey.
2. Enter a name: e.g. **Gripper_1**. If it exists, the display text is immediately displayed from patm_xx.ini.
3. Select the type: Transfer location, **gripper**, loader, spindle
4. The buffer location is created by pressing "OK", the location number and the index are internally 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.



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

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 226)

6.1.3 Loading points and loading locations

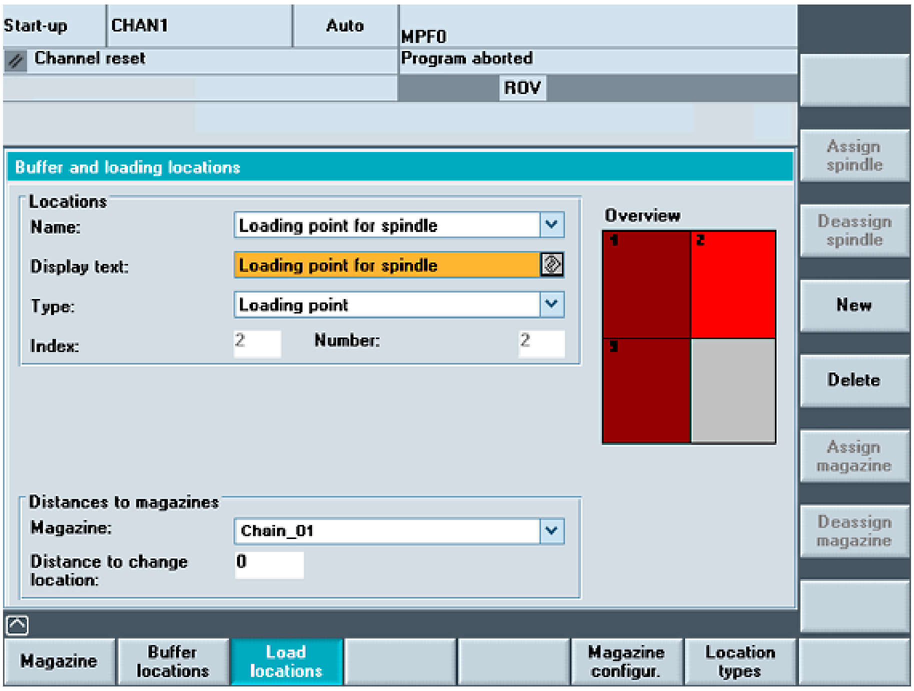
Loading locations

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:

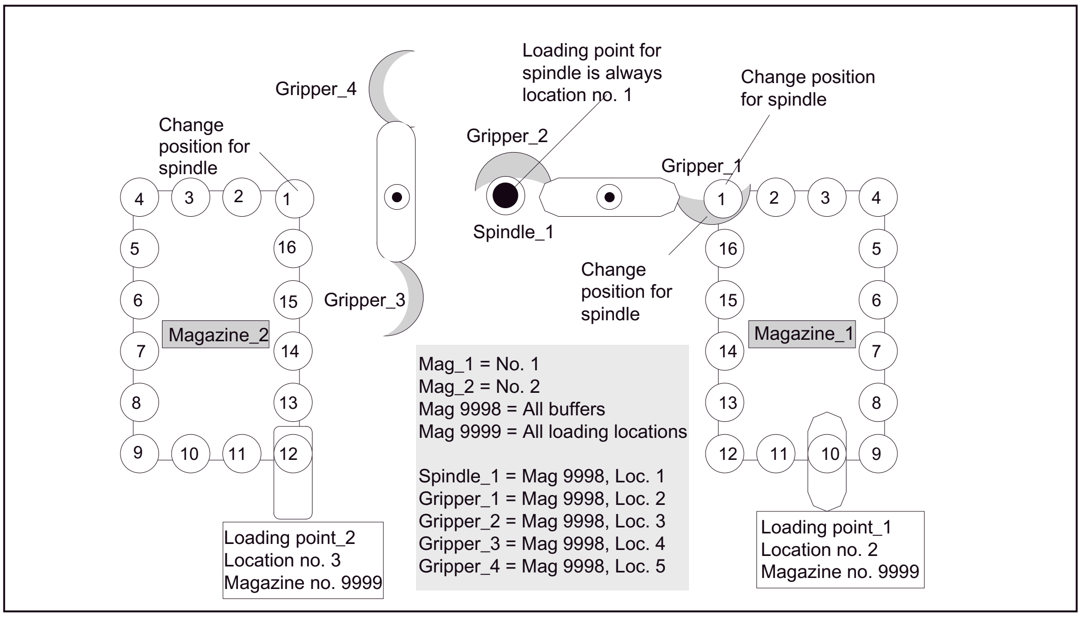
1. Press the "New" softkey.
2. Enter name: e.g. Loading_1. If it exists, the display text is immediately displayed from patm_xx.ini.
3. Select the type: e. g. Loading point
4. "OK" creates the loading point. The location number and the index are internally assigned and incremented.

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.



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

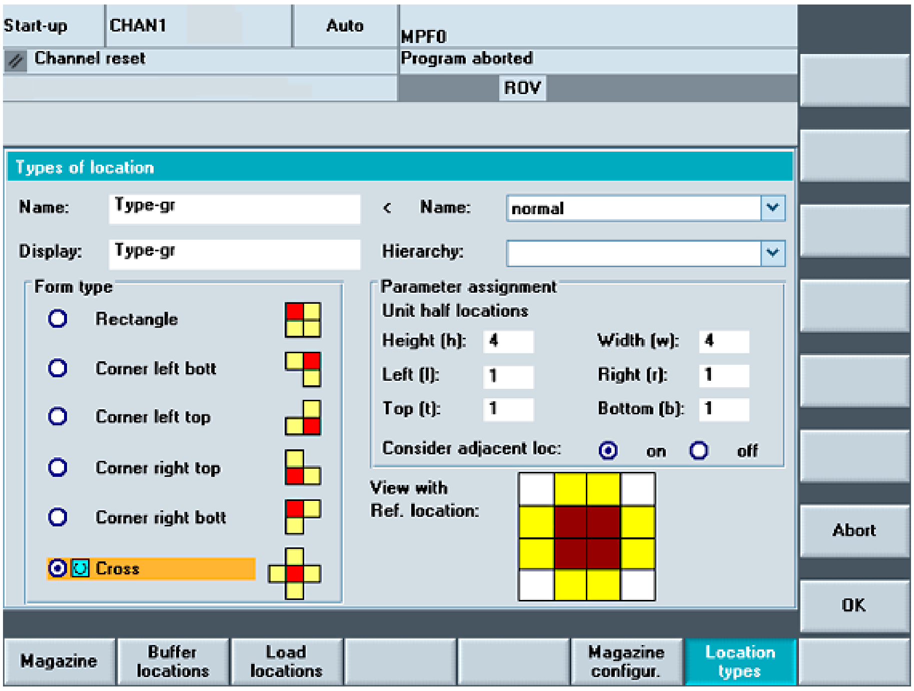
Figure 6-5 Example of a machine with 2 magazines and 3 loading locations

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".

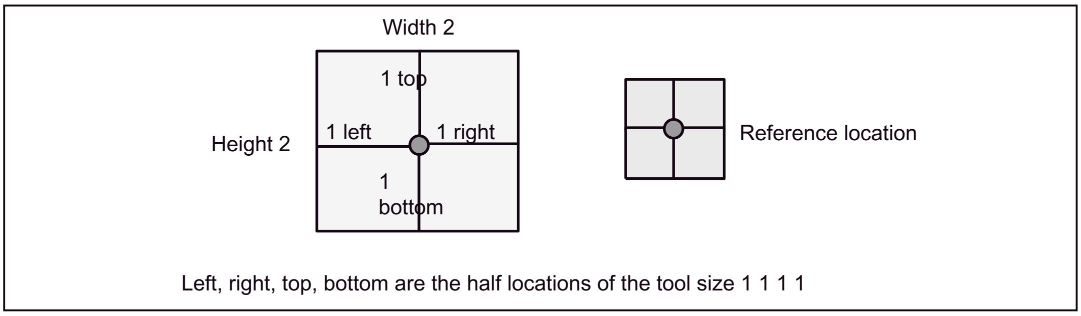

Width 2
1 top
Height 2
1 left
1 right
1
bottom
Reference location
Left, right, top, bottom are the half locations of the tool size 1 1 1 1

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.

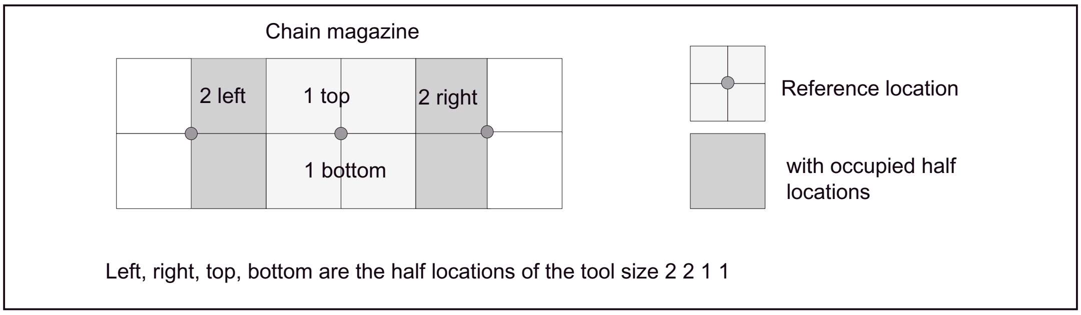

Chain magazine
2 left
1 top
2 right
1 bottom
Reference location
with occupied half locations
Left, right, top, bottom are the half locations of the tool size 2 2 1 1

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.

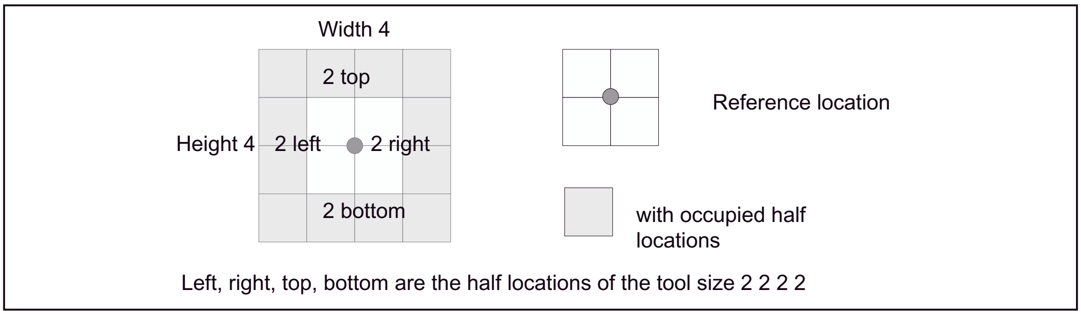

Width 4
2 top
Height 4
2 left
2 right
2 bottom
Reference location
with occupied half locations
Left, right, top, bottom are the half locations of the tool size 2 2 2 2

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.

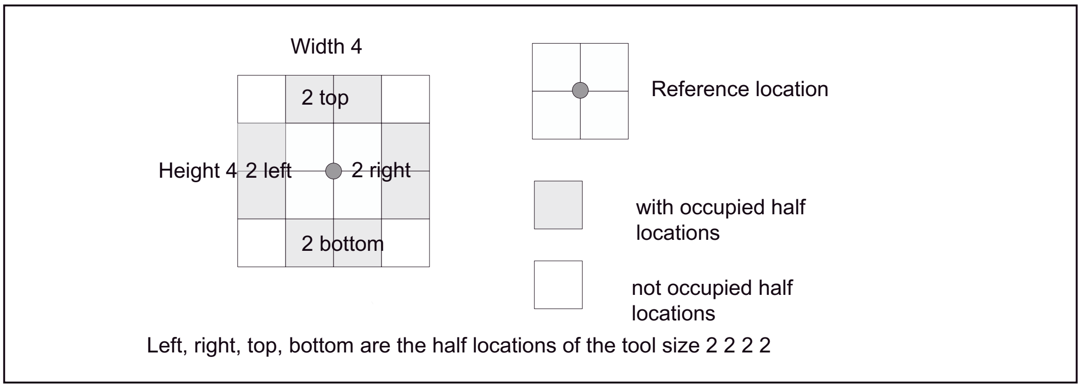

Width 4
2 top
Height 4
2 left
2 right
2 bottom
Reference location
with occupied half locations
not occupied half locations
Left, right, top, bottom are the half locations of the tool size 2 2 2 2

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:

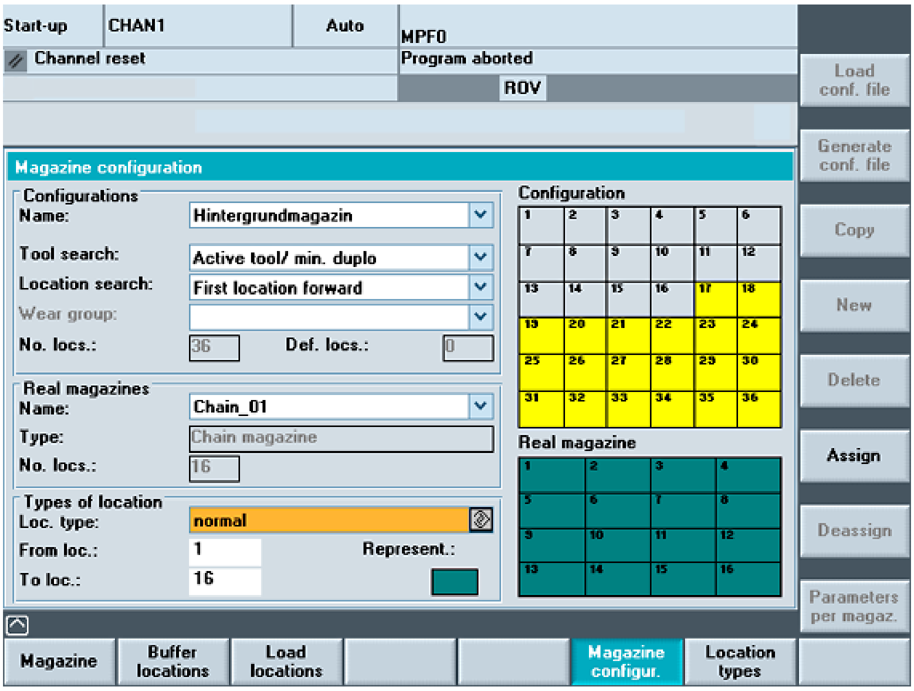
- You can remove the hierarchy selected in the "Hierarchy" field by pressing the "Delete hierarchy" softkey.

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 Represent.:
To loc.: 16

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

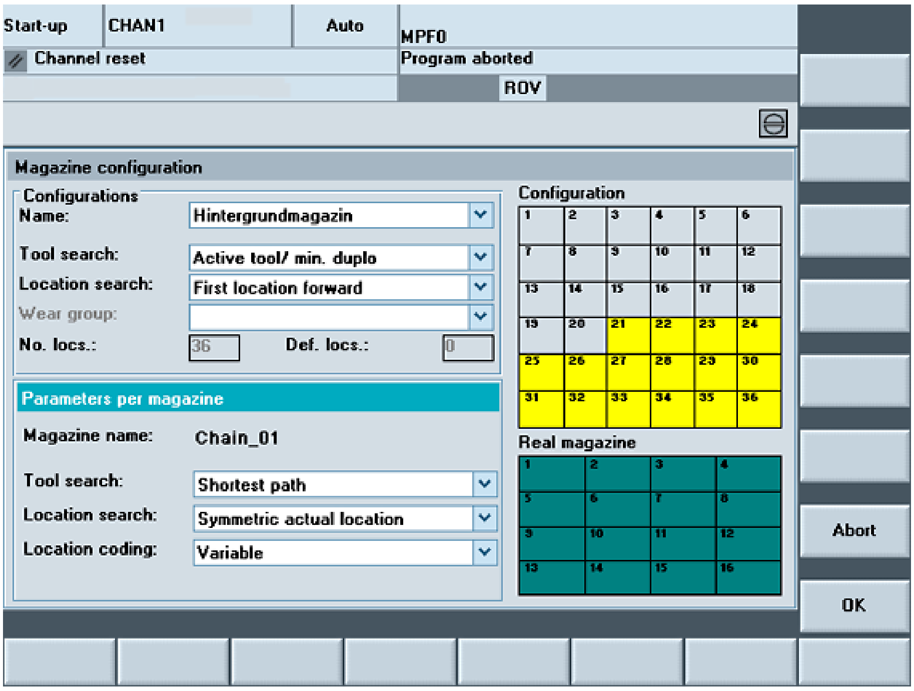
Name:	Name of the selected real magazine
Type:	Type of magazine
No. of Locations:	Number of locations for the selected magazine

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.

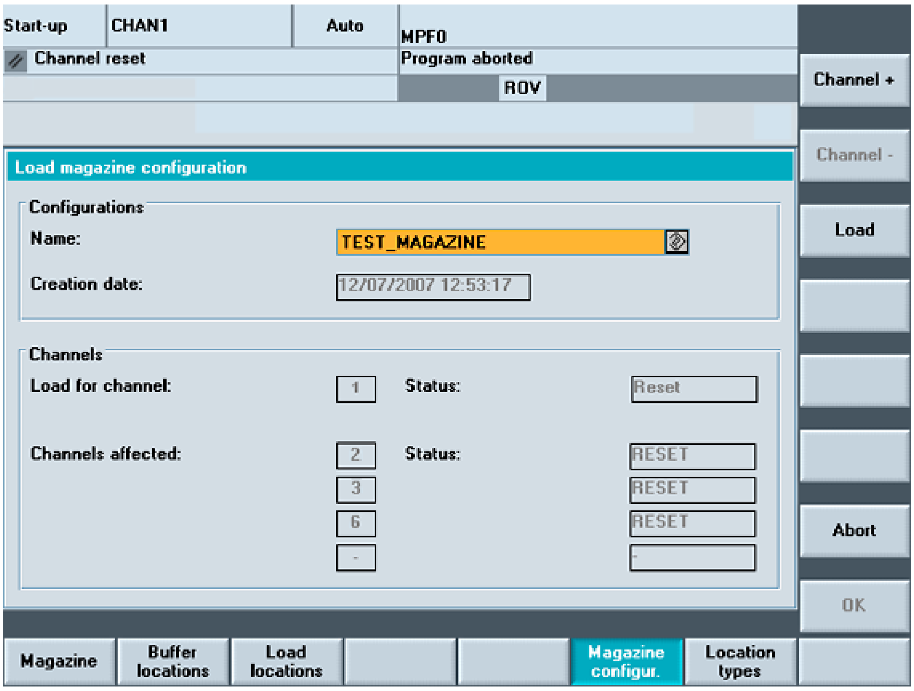
Delete magazine configuration:

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 246)

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).
; gesucht. Die Zeichenfolge "gr" in "patm_gr.ini" hängt von der
; 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
; zulässig sein. Falls 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.
; erzeugt. Das BTSS-Item muss in "" eingeschlossen sein,
; 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]

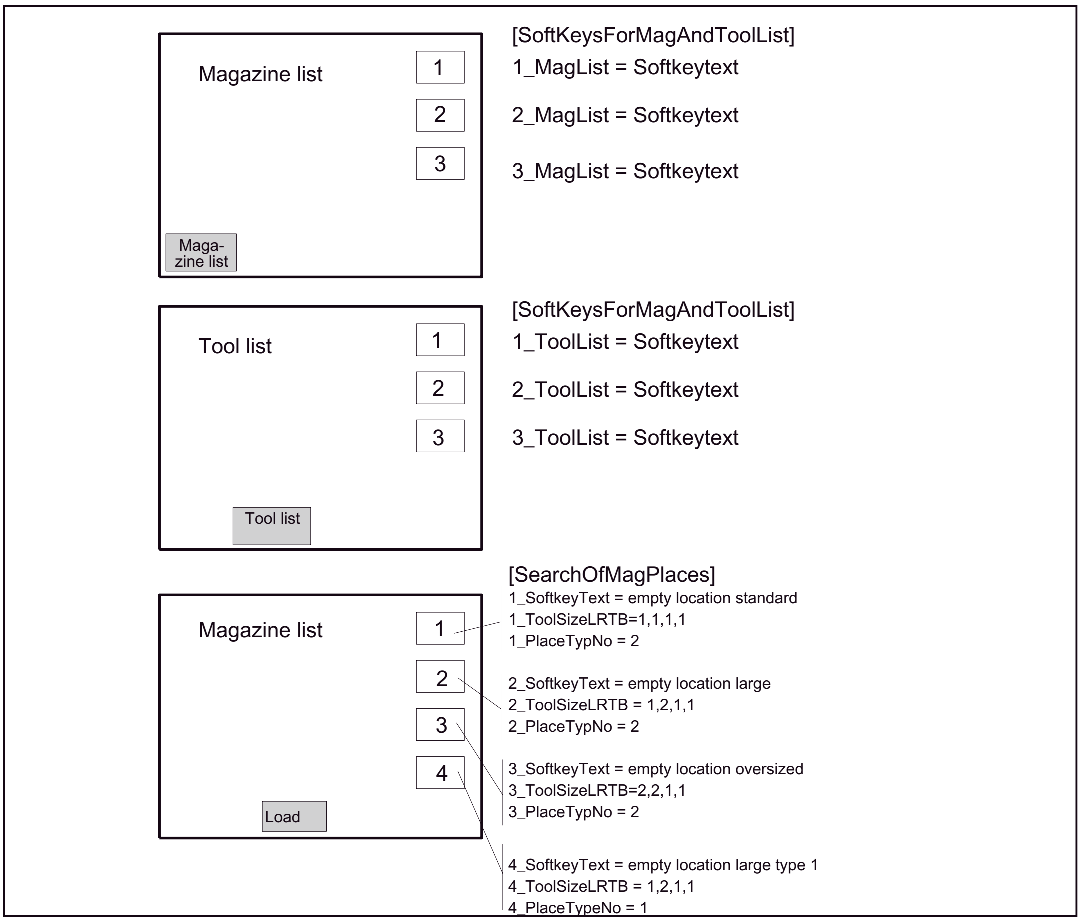
```
; Definition of shortcut softkeys for up to 5 favorite magazines
; per TOA, evaluated if section "[GeneralSettingsForMagAndToolList]"
; entry "MagListMagSelectSoftkey=SelectMag" is set.
; This helps to prevent users from excessive use of "magazine +"
; and "magazine -" softkeys, if a lot of magazines 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

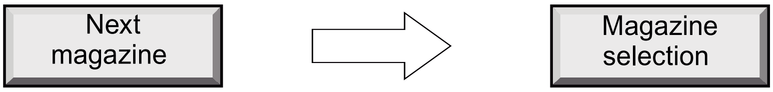
Special characters

Special characters such as ü, ä, ö, ß are entered in ANSI code in order for them to be displayed.

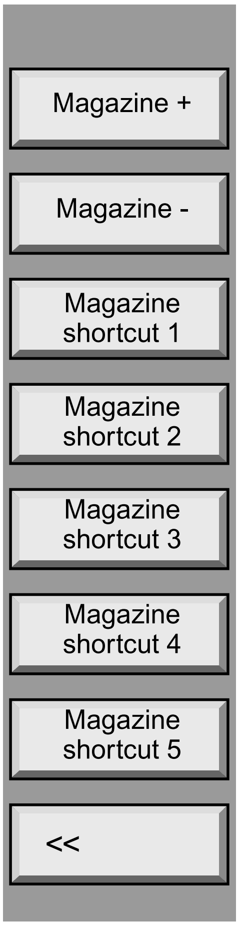
Optional selection of magazines

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

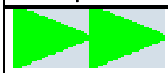
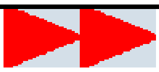
Parameter settings for bitmaps in the list

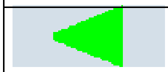
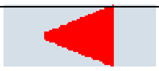
The display of the active tool, the programmed tool and the current location in the magazine list can be freely parameterized, i.e. bitmaps can be inserted in the parameterizable columns of individual lists. The bitmaps can be customized or created from scratch. The display is activated in paramtm.ini.

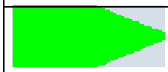
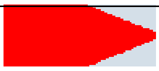
The bitmaps described below are located in the "hmi_adv" directory.

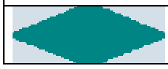
Default bitmap settings

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](I)"

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 246)

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
C35	Double	Prewarning limit for wear	$TC_MOP5
C36	Double	Wear	$TC_MOP6
C37	Double	Setpoint wear	$TC_MOP15

Dialog variable	Data type	Designation	Assignment $TC...
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       '  Continue reading Block 1 until the count
BItem1                  '  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>
```

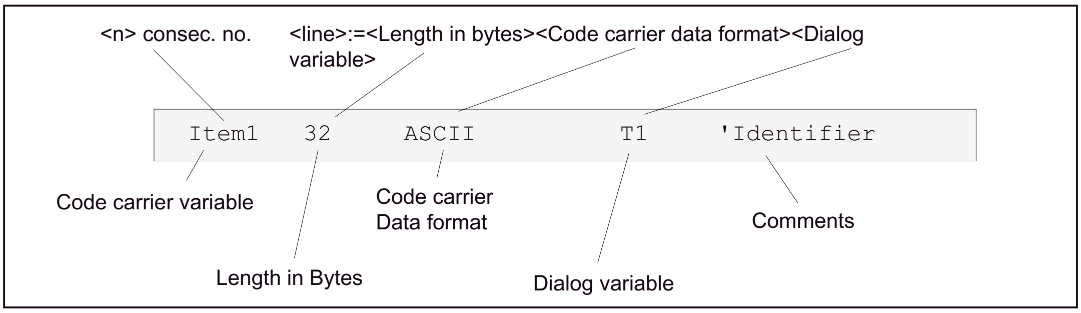

<n> consec. no.
<line>:=<Length in bytes><Code carrier data format><Dialog variable>
Item1   32   ASCII   T1   'Identifier
Code carrier variable
Code carrier Data format
Comments
Length in Bytes
Dialog variable

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
Bitem2	1	BCD	C4	' Cutting edge position
' Tool length compensation				

Code carrier variable	Length (bytes)	Data format	Dialog variable	Comments
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:

```
626F687265725F312020000111116E6F726D616C2020202001D00010E3D000000502
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
`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)		

A

List of the INI Files

Please observe:

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 262)

AEDITOR.INI (Page 263)

DINO.INI (Page 264)

DG.INI (Page 265)

DGOVW.INI (Page 266)

DH.INI (Page 267)

DPDH.INI (Page 267)

DPSIM.INI (Page 267)

HEADER.INI (Page 268)

HMIDESK.INI (Page 272)

IB.INI (Page 273)

IF.INI (Page 274)

KEYS.INI (Page 275)

LOGDRIVE.INI (Page 275)

MACHINE.INI (Page 278)

MBDDE.INI (Page 281)

MMC.INI (Page 285)

NETNAMES.INI (Page 290)

OEMFRAME.INI (Page 293)

PARAM.INI (Page 294)

PARAMTM.INI (Page 295)

REGIE.INI (Page 295)

SEDITOR.INI (Page 297)

SIMTOGEO.INI (Page 304)

TASKCONF.INI (Page 306)

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]
MemFactor=f ; f stands for 8 MB units. Permissible values are
1<=f<=31
 ; corresponding to 8 MB to 248 MB
```

Editor access authorizations are influenced in the Section [ACCESSLEVEL]:

```
[ACCESSLEVEL]
GlobalReplace=6
SettingHiddenCheck=7
Enable_Simulation = 7
; in the Editor, the Simulation softkey can be hidden.

[MMC_Wizard]
AEDITOR=AEDITOR.COM

[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, 31 = 248 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
; the application cannot override this default factor
MemNCFactor=2
; Number of steps (input lines!), that the ASCII editor
; can reset (Undo)/restore (Redo).
; Default=0: No Undo/Redo steps can be activated
; > 0 max. 32767 only applies if the setting for UndoHandling does
not equal 2.
NrOfUndoSteps=10
; Mode for Undo/Redo handling (synchronization with application):
; UndoHandling=0 Automatic: Application responds immediately to
Ctrl+Z, CTRL+Y, Default
```

```
; UndoHandling=1 active application is informed that Ctrl+Z, CTRL+Y
has come, application-specific response
; UndoHandling=2 no Undo/Redo functionality
UndoHandling=0
; Also applies to NC files
; EnableUndoOnNc=1 NC file is also processed with Undo/Redo
```

With this mode it takes longer to prepare the NC file for processing in the HMI.

```
; EnableUndoOnNc=0 NC file is not processed with Undo/Redo□;
(Default)
EnableUndoOnNc=0

[STEPS]
; maximum number of blocks to search for a step, 0 == no limit
MaxNrOfBlocks=1500

[ACCESSLEVEL]
GlobalReplace=6
SettingHiddenCheck=7
Enable_Simulation=7 ; Protection level for softkey Simulation in
Editor
```

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 user-specific signals in 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 created explicitly in the directory mmc2 or addon, oem, 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 language abbreviation. The file contains the language-specific texts for user-specific signals in the service overview. The file must be created explicitly in the mmc2 or 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.
; Example:
$T1001="Signal s"
```

For details on configuring and for an example, refer to:

See also

Supplementing service displays on a user-specific basis (Page 34)

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 ModeGroup ready
 3=CondStop.bmp        ;EMERGENCY STOP active
 4=CondStop.bmp        ;Alarm active with stop
 5=CondStop.bmp        ;M0 / M1 active
```

```
6=CondStop.bmp         ;Block ended in single block mode
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 feed override
18=CondStop.bmp        ;Error in NC block
19=CondStop.bmp        ;Waiting for external NC blocks
20=CondWait.bmp        ;Waiting for synchronous action
21=CondWait.bmp        ;Block search active
22=CondStop.bmp        ;No spindle enable
23=CondStop.bmp        ;Axis feed override 0
24=CondWait.bmp        ;Waiting for tool change acknowledgment
25=CondWait.bmp        ;Gear stage 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 replacement active (block change
                       inhibited because axis replacement 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 replacement: Superimposed
                       motion
41=CondWait.bmp        ;AXCT: Axis replacement 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 return initiated
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     ;Hold at cycle end M0
64=CondWait.bmp     ;Hold at cycle end M1
65=CondWait.bmp     ;Wait: Axis is at endstop
66=CondWait.bmp     ;"Master-Slave changeover active"
67=CondWait.bmp     ;"Axis exchange: Axis single axis"
68=CondWait.bmp
69=CondWait.bmp     ;"Synchronism: Synchronous spindle"
70=CondWait.bmp     ;"Switch-off position synch. spindle"
71=CondWait.bmp     ;"Wait for enable of the transformer axis"
72=CondWait.bmp
73=CondWait.bmp
```

```
[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 limited 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 programming device or standard 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 "Start-up" operating area can be set individually by entering the protection level. Enter the required protection level in section [ACCESSLEVEL]. The operator languages are defined in Section [LANGUAGE].

```
[ACCESSLEVEL]
RESET =2 ;NCK reset NCK reset
CHG_SCL_SYS=5 ;change scaling system, system of dimensions,
inch/metric
MASCHDAT =4 ;Machine data
DRIVE_CFG=2 ;Drive configuration
BOOTFILE=2 ;Boot file
MOT_CTRL=2 ;Motor/Valve selection
VIEW=2 ;User views + Display options

NC_ADDR=2 ;NCK address
SETUP_SWITCH=2;commissioning switch
PLC_STATE=3 ;plc state
PLC_DATE=3 ;plc date
PLC-CHANGE=2 ;PLC status :
 ;Sk Change Value (change)
 ;Sk Reset Value (undo change)
 ;Sk Commit Value (accept change)
 Interlocked by manufacturer
DRIVE_SERVO=2 ;Drives/Servo
SET_COL=3 ;colors
SYST_SETTINGS=4 ;system settings
SET_LANG=2 ;Language
SET_BTSS=2 ;Operator panel
OPTION=2 ;Options
EDIT=1 ;Editor
DOS=1 ;DOS shell
TOOL_MNG=3 tool management
FILE=7 ;File functions
PASSWD=3 ;Set/Delete password Change and delete password
CHG_PASSWD=7 ;SK "password ..."
CHANGE_LANG=7 ;change language

[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
```

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.

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.

```
[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 exist.
 ACTTCU:\\FRONT always can be set
; all files LOGDRIVE.INI of the <SEARCH PATH> are read and the
contents is 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 to a usb devices 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 to a usb devices 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.

Note

The logical drives can also be set-up using the operator interface of HMI-Advanced under "Start-up" → "HMI" → "Logic drives".

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_L
4_INT
;ADDRESS1=192.168.214.1,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_
L4_INT
;ADDRESS2=192.168.214.1,LINE=10,NAME=/CP,SAP=0501,PROFILE=CLT1__CP_L
4_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 at the operator interface HMI 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
; originally were developed (typically: 640x480)
; BaseScreen - default: 640x480
BaseScreen=640x480
```

If a mouse is connected to the control, it is positioned as standard in 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.

The OPI settings are the display machine data that can also also be set for a specific technology using the operator interface of the HMI in the operator area "Start-up" → "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 using the operator interface of the HMI in the operator area "Start-up" → "HMI" → "System settings":

```
; ActionLog (action log) parameters


[ActionLog]
; CommitLatency tells the maximum number of seconds that elapse 
before 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 **multichannel 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 297)

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 50)

Configuring a double-channel display (Page 53)

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 175)

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: 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 an 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 are 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)
; '#' : seperator: text before is searched in one
; channel, text behind in all others.
; It isn't possible to synchronize this kind of
; synchronization marks !!!
; '*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 a 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 string. 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 number, these do not have to be in the same sequence,
 the separate 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 then 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 subblock 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 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" if the version. Generally, this is the text that is found in the NC program after the path of the COM file (NC1, NC2, ...).

`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 subblock or a main block.

```
1: Sub-block
0: Main block
```

References: 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

Abbreviation	Significance
MSD	Main Spindle Drive
Hardware	Hardware
IBN	Commissioning
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: 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

Abbreviation	Significance
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

Index